

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Máster

**Caracterización experimental del
comportamiento de TCP sobre enlaces
802.11n de 40 MHz**

(Experimental characterization of TCP performance on
802.11n 40 MHz-width channels)

Para acceder al Título de

***Máster Universitario en
Ingeniería de Telecomunicación***

Autor: Rodrigo Gañán de Andrés

Julio - 2021



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE MASTER

Realizado por: Rodrigo Gañán de Andrés
Director del TFM: Marta García Arranz

Título: “Caracterización experimental del comportamiento de TCP sobre enlaces 802.11n de 40 MHz”

Title: “Experimental characterization of TCP performance on 802.11n 40 MHz-width channels”

Presentado a examen el día: 27/07/2021

para acceder al Título de

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): Ramón Agüero Calvo

Secretario (Apellidos, Nombre): Beatriz Aja Abelán

Vocal (Apellidos, Nombre): Alberto Eloy García Gutierrez

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFM
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Máster Nº
(a asignar por Secretaría)

Resumen

Dentro del Grupo de Ingeniería Telemática (GIT) se han venido realizando numerosos trabajos en el campo de la caracterización del rendimiento de los estándares IEEE 802.11 para redes de área local inalámbrica (WLAN). Hasta la fecha se disponía de un conocimiento profundo de cómo era dicho comportamiento para las primeras especificaciones a/b/g del estándar; sin embargo, la norma 802.11n, aprobada en 2009 y de uso mayoritario actualmente, introduce importantes mejoras, tanto en la capa física como en la capa MAC, por lo que se ha hecho necesario ampliar dicha caracterización. Una de estas mejoras consiste en la técnica denominada "Channel Bonding", que básicamente es la unión de dos canales de 20 MHz en uno de 40 MHz. Para establecer las diferencias que presenta un canal de 40 MHz frente a un canal de 20 MHz, se han realizado una serie de medidas para caracterizar el rendimiento de una red Wi-Fi utilizando tráfico TCP. Las distintas medidas vendrán condicionadas por el algoritmo de control de congestión de TCP utilizado (New Reno o Cubic), la distancia entre cliente (emisor) y servidor (receptor) dentro de una misma red Wi-Fi, y la caracterización del rendimiento de la red dependiendo de si hay o no tráfico de otras fuentes dentro de la misma red Wi-Fi. Este trabajo es una continuación del TFG "Caracterización experimental del rendimiento del estándar 802.11n en un entorno doméstico" realizado también por el alumno Rodrigo Gañán de Andrés.

Abstract

The Telematics Engineering Group (GIT) has been doing a lot of work in the field of performance characterization of IEEE 802.11 standards for wireless local area networks (WLAN). However, the 802.11n standard, approved in 2009 and currently in widespread use, introduces significant improvements in both the physical layer and the MAC layer, making it necessary to extend this characterization. One of these improvements consists of the technique called "Channel Bonding", which is basically the union of two 20 MHz channels into one 40 MHz channel. To establish the differences between a 40 MHz channel and a 20 MHz channel, a series of measurements have been carried out to characterize the performance of a Wi-Fi network using TCP traffic. The different measurements will be conditioned by the TCP congestion control algorithm used (New Reno or Cubic), the distance between client (sender) and server (receiver) within the same Wi-Fi network, and the characterization of the performance of the network depending on whether or not there is traffic from other sources within the same Wi-Fi network. This work is a continuation of the TFG "Experimental characterization of 802.11n performance in home networks" also carried out by the student Rodrigo Gañán de Andrés.

Índice general

Capítulo 1. Introducción.....	1
1.1. Presentación y objetivos.....	1
1.2. Estructura de la memoria	2
Capítulo 2. Aspectos Teóricos.....	3
2.1. Conceptos básicos del estándar IEEE 802.11n	3
2.2. Agregación de tramas en el estándar 802.11n	6
2.2.1. A-MSDU.....	7
2.2.2. A-MPDU	7
2.3 Block ACK	8
2.4. Protocolo TCP.....	9
2.4.1. Cubic	10
2.4.2. New Reno.....	11
Capítulo 3. Aspectos prácticos.....	12
3.1. Descripción de la plataforma de medidas	12
3.2. Escenario.....	14
3.2.1. Escenario de menor distancia	14
3.2.2. Escenario de distancia media.....	16
3.2.3. Escenario de mayor distancia.....	17
3.3. Tráfico interferente de otras redes	18
3.4. Herramientas empleadas en la campaña de medidas	22
3.4.1. Herramienta de medida – Iperf3.....	22
3.4.2. Programas para análisis, obtención y representación de datos	22
Capítulo 4. Realización de las medidas	24
4.1. Creación de la red 802.11n	24
4.2. Captura de datos	27
4.3. Obtención datos con C	33
4.3.1. Lectura y extracción de datos	34
4.3.2. Escritura de resultados	37
4.4. Matlab	40
Capítulo 5. Análisis de los resultados.....	44
5.1. Beacon de la red Wi-Fi.....	44
5.2. Cálculo del throughput teórico	46
5.3. Análisis de los resultados de las medidas	54
5.3.1. Escenario de menor distancia	55

5.3.2. Escenario de distancia media.....	69
5.3.3. Escenario de mayor distancia.....	82
5.4. Retransmisiones MAC.....	95
Capítulo 6. Conclusiones y líneas futuras de trabajo	96
Bibliografía	99
Anexo A. Código C “ProcesaTCP.c”	100
Anexo B. Código C “ProcesaCTS.c”	102
Anexo C. Código Matlab “dibuja.m”	103
Anexo D. Código Matlab “dibujaCTS.m”	104

Índice de Figuras

Figura 1.1. Red Wi-Fi en modo Ad-hoc	1
Figura 2.1. Channel bonding: representación canal de 40 MHz que aglutina los canales 5 y 9 y cuya anchura es la zona sombreada mientras que los demás semicírculos representan canales de 20 MHz	3
Figura 2.2. Dependiendo del número de antenas del punto de acceso y del dispositivo se decidirá el número de spatial streams	4
Figura 2.3. Características obligatorias frente a las características opcionales que puede presentar el estándar 802.11n, divididas según el beneficio que ofrecen: mejor rendimiento, mejor fiabilidad o inter-operabilidad.....	6
Figura 2.4. Funcionamiento agregación A-MSDU	7
Figura 2.5. Funcionamiento agregación A-MPDU.....	8
Figura 2.6. Reconocimiento simple vs Reconocimiento en Bloque [5].....	9
Figura 2.7. Evolución ventana de congestión en TCP Cubic	11
Figura 3.1. Imagen del adaptador inalámbrico TP-LINK modelo TL-WN821N	13
Figura 3.2. Respuesta al comando lsusb	13
Figura 3.3. Respuesta al comando “dmesg grep ‘ath’” donde ath representa el código de driver de Atheros	13
Figura 3.4. Distribución de los ordenadores del laboratorio 128	14
Figura 3.5. Plano aproximado para el escenario de menor distancia	15
Figura 3.6. Escenario de menor distancia	15
Figura 3.7. Plano para el escenario de menor distancia con tráfico interferente	15
Figura 3.8. Plano aproximado para el escenario distancia media	16
Figura 3.9. Escenario de distancia media	16
Figura 3.10. Plano para el escenario distancia de media con tráfico interferente.....	17
Figura 3.11. Plano aproximado para el escenario de mayor distancia	17
Figura 3.12. Escenario de mayor distancia.....	17
Figura 3.13. Plano para el escenario de mayor distancia con tráfico interferente.....	18
Figura 3.14. Datos del entorno Wi-Fi para una red Wi-Fi de 20 MHz.....	19
Figura 3.15. Representación potencia de señal recibida y otros datos de una red Wi-Fi de 20 MHz	19
Figura 3.16. Representación gráfica del espectro de 2.4 GHz para un canal de 20 MHz.....	20
Figura 3.17. Datos del entorno para una red Ad-Hoc de 40 MHz.....	21
Figura 3.18. Representación potencia de señal recibida y otros datos de una red Wi-Fi de 40 MHz.....	21
Figura 3.19. Representación gráfica del espectro de 2.4 GHz para un canal de 40 MHz.....	22
Figura 3.20. Iconos de Wireshark, Dev-C++ y Matlab en este orden.....	23

Figura 4.1. Resultados comando iwconfig.....	25
Figura 4.2. Adaptador USB Wi-Fi trabajando en modo Ad-Hoc	25
Figura 4.3. Resultados de iwconfig después de crear una red Wi-Fi Ad-Hoc.....	26
Figura 4.4. Comprobación Dirección IP	26
Figura 4.5. Resultado Ping	26
Figura 4.6. Información del dispositivo y la red Wi-Fi a la que esté conectada	27
Figura 4.7. Resultado comando iwconfig para un dispositivo en modo monitor.....	28
Figura 4.8. Información de un dispositivo en modo monitor.....	28
Figura 4.9. Resultado iperf3 (1)	30
Figura 4.10. Resultados finales iperf3 (2).....	30
Figura 4.11. Resultados finales iperf3 del servidor (3).....	31
Figura 4.12. Filtros empleados para obtener los segmentos TCP.....	31
Figura 4.13. Imagen con las ventanas descritas desplegadas de un paquete TCP.	32
Figura 4.14. Filtros empleados para obtener tramas CTS	32
Figura 4.15. Imagen con las ventanas descritas desplegadas de una trama CTS	33
Figura 4.16. Ciclo While para la lectura del archivo.....	34
Figura 4.17. Resumen inicial de un segmento TCP	34
Figura 4.18. Lectura del tamaño del segmento	35
Figura 4.19. Lectura de la velocidad de transmisión.....	35
Figura 4.20. Lectura de la velocidad de transmisión.....	36
Figura 4.21. Lectura del identificador IP y comprobación si el segmento es una retransmisión TCP o un segmento TCP fuera de orden	36
Figura 4.22. Resumen inicial de una trama CTS.....	37
Figura 4.23. Lectura del instante de tiempo en que se captura la trama	37
Figura 4.24. Lectura tiempo otorgado al transmisor	37
Figura 4.25. Escritura de los valores de potencia de señal recibida de cada ráfaga A-MPDU	38
Figura 4.26. Visualización del archivo "SolSen.txt"	38
Figura 4.27. Escritura de los valores de velocidad de transmisión de cada ráfaga A-MPDU	38
Figura 4.28. Visualización del archivo "SolRate.txt"	39
Figura 4.29. Escritura del instante de tiempo en segundos en que se recibió la trama CTS y la cantidad de tiempo que se le ha otorgado al transmisor para enviar información en microsegundos	39
Figura 4.30. Visualización del archivo "CTSPprocesado.txt"	40
Figura 4.31. Lectura datos del archivo SolSen.txt.....	40
Figura 4.32. Código para la creación de una gráfica que represente el RSSI	41
Figura 4.33. Representación gráfica del RSSI	41
Figura 4.34. Lectura datos del archivo SolRate.txt	41

Figura 4.35. Código para la creación de una gráfica que represente las velocidades de transmisión de datos.....	42
Figura 4.36. Representación gráfica de las velocidades de transmisión de datos	42
Figura 4.37. Lectura datos del archivo CTSprocesado.txt	42
Figura 4.38. Código para la creación de una gráfica que represente el instante de llegada de una trama CTS y cuánto tiempo le ha sido otorgado en esa trama CTS al transmisor para enviar información.	43
Figura 4.39. Representación gráfica del tiempo otorgado para enviar datos al transmisor	43
Figura 5.1. Tramas Beacon de la red Wi-Fi Ad-Hoc creada en Wireshark.....	44
Figura 5.2. Información red Wi-Fi Ad-Hoc creada recogida en la trama de Beacon.....	45
Figura 5.3. Características opcionales de 802.11n de la red Wi-Fi creada recogida en la trama de Beacon.....	45
Figura 5.4. Datos sobre el canal de una red Wi-Fi recogida en la trama de Beacon	45
Figura 5.5. Nº de spatial streams de la red Wi-Fi Ad-Hoc de trabajo	46
Figura 5.6. Esquema de acceso basado en intercambio RTS/CTS y Block ACK.....	46
Figura 5.7. Evolución throughput teórico para un canal de 20 MHz	53
Figura 5.8. Evolución throughput teórico para un canal de 40 MHz	54
Figura 5.9. Evolución del throughput durante la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic.....	56
Figura 5.10. Evolución de la ventana de congestión durante la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic	56
Figura 5.11. Potencia de señal recibida en el receptor en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic	57
Figura 5.12. Velocidad de transmisión en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic.....	57
Figura 5.13. Tiempo disponible para transmitir del transmisor en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic	58
Figuras 5.14. Resultados de la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno	59
Figura 5.15. Número de retransmisiones en cada instante de tiempo en la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic	60
Figuras 5.16. Resultados de la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic.....	61
Figuras 5.17. Resultados de la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando New Reno	63
Figuras 5.18. Resultados de la medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic.....	64
Figuras 5.19. Resultados de la medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno	66

Figuras 5.20. Resultados de la medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando Cubic.....	67
Figuras 5.21. Resultados de la medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando New Reno	69
Figuras 5.22. Resultados de la medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando Cubic.....	70
Figuras 5.23. Resultados de la medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando New Reno	72
Figuras 5.24. Resultados de la medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando Cubic.....	73
Figuras 5.25. Resultados de la medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando New Reno	75
Figuras 5.26. Resultados de la medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando Cubic.....	76
Figuras 5.27. Resultados de la medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando New Reno	78
Figuras 5.28. Resultados de la medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando Cubic.....	80
Figuras 5.29. Resultados de la medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando New Reno	81
Figuras 5.30. Resultados de la medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic.....	83
Figuras 5.31. Resultados de la medida del Escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno	85
Figura 5.32. Evolución del Throughput durante la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic.....	86
Figuras 5.33. Resultados de la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic.....	87
Figuras 5.34. Resultados de la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando New Reno	88
Figuras 5.35. Resultados de la medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic.....	90
Figuras 5.36. Resultados de la medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno	91
Figuras 5.37. Resultados de la medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando Cubic.....	93
Figuras 5.38. Resultados de la medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando New Reno	94

Índice de tablas

Tabla 2.1. MCSs y velocidades de transmisión del estándar 802.11n	5
Tabla 5.1. Parámetros de la capa física y MAC de distintos estándares 802.11 [9]	47
Tabla 5.2. Valor de <i>NLTF</i> según el número de Spatial Streams [9]	48
Tabla 5.3. Tiempos reales de las tramas MPDU contenidas dentro de una A-MPDU	50
Tabla 5.4. Valores de los parámetros de capa física y MAC	51
Tabla 5.5. Throughput teórico vs. Throughput real	52
Tabla 5.6. Resultados medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic	55
Tabla 5.7. Resultados medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno	58
Tabla 5.8. Resultados medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic	60
Tabla 5.9. Resultados medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando New Reno	62
Tabla 5.10. Resultados medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic	63
Tabla 5.11. Resultados medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno	65
Tabla 5.12. Resultados medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando Cubic	66
Tabla 5.13. Resultados medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando New Reno	68
Tabla 5.14. Resultados medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando Cubic	69
Tabla 5.15. Resultados medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando New Reno	71
Tabla 5.16. Resultados medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando Cubic	72
Tabla 5.17. Resultados medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando New Reno	74
Tabla 5.18. Resultados medida del Escenario distancia media Canal de 40 MHz sin tráfico interferente utilizando Cubic	75
Tabla 5.19. Resultados medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando New Reno	77
Tabla 5.20. Resultados medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando Cubic	79
Tabla 5.21. Resultados medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando New Reno	80

Tabla 5.22. Resultados medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic.....	82
Tabla 5.23. Resultados medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno	84
Tabla 5.24. Resultados medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic.....	85
Tabla 5.25. Resultados medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando New Reno	87
Tabla 5.26. Resultados medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic.....	89
Tabla 5.27. Resultados medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno	90
Tabla 5.28. Resultados medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando Cubic.....	92
Tabla 5.29. Resultados medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando New Reno	93
Tabla 5.30. Porcentaje de retransmisiones a nivel de capa MAC en cada medida	95

Acrónimos

IEEE: Institute of Electrical and Electronics Engineers

IP: Internet protocol

RSSI: Received Signal Strength Indicator

SOC: System On a Chip

USB: Universal Serial Bus

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

Wi-Fi: Wireless Fidelity

MPDU: MAC Protocol Data Unit

MSDU: MAC Service Data Unit

A-MPDU: Aggregated MPDU

A-MSDU: Aggregated MSDU

MIMO: Multiple-input Multiple-output

CPU: Central Processing Unit

LDPC: Low Density Parity Check

STBC: Space–time block coding

TxBF: Transmit Beamforming

MCS: Modulating and Coding Scheme

IBSS: Independent Basic Service Set

TxOP: Transmission Opportunity

Palabras clave

Rendimiento, Channel Bonding, Iperf3, Throughput, Linux, Wi-Fi, Ad-Hoc, medidas, 802.11n, Retransmisión, Agregación de tramas, Cubic, New Reno, Throughput, Ventana de Congestión y Mecanismo de transmisión (Rate adaptation).

Capítulo 1. Introducción

1.1. Presentación y objetivos

Con el paso del tiempo, el tráfico que mueve una red Wi-Fi se ha ido incrementando exponencialmente, y no solamente eso, sino el incremento del número de redes Wi-Fi, las cuales muchas veces comparten canal. Esto ha provocado una saturación del canal y la búsqueda de soluciones efectivas para armonizar esos intercambios de datos presentes en una red Wi-Fi. Con el incremento de las tasas de transmisión, como es de esperar, se ha incrementado la velocidad de acceso a Internet, pero también este incremento está ligado a una mayor pérdida de paquetes debido a la saturación de la red. Para ello, se han desarrollado una serie de algoritmos para el control de la congestión, para conseguir transmitir a la mayor velocidad posible, sin sufrir pérdidas. Estos algoritmos difieren unos de otros e incluso hay algunos destinados a situaciones específicas, como puede ser una red Wi-Fi con muchas pérdidas.

En este trabajo se probará el funcionamiento de dos algoritmos de control de congestión, Cubic y New Reno, que son muy utilizados actualmente, y se comprobará su eficacia. Estas pruebas se realizarán en un entorno controlado en la Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Para ello se utilizará los ordenadores del laboratorio 128 de la primera planta. Con estos ordenadores se simulará una red Wi-Fi sobre la cual se realizarán las diferentes pruebas.

La red Wi-Fi tendrá una topología Ad-Hoc, donde los clientes de la red (ordenadores) se comunican entre ellos directamente sin la necesidad de un punto de acceso como puede ser un router. Los ordenadores se conectarán a esta red Wi-Fi utilizando adaptadores USB Wi-Fi.

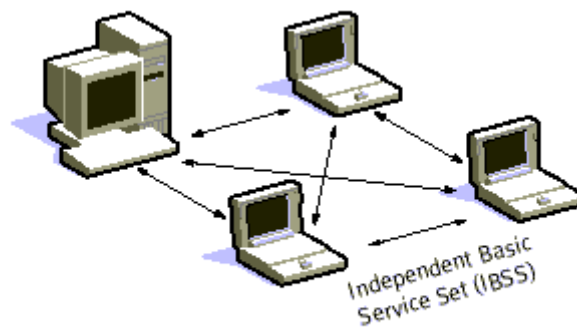


Figura 1.1. Red Wi-Fi en modo Ad-hoc

Las medidas se realizarán en tres escenarios, que vendrán determinados por la distancia entre el transmisor y el receptor: menor distancia, distancia media y mayor distancia. Al realizar estas pruebas se observará como influye la distancia en la transmisión de datos en una red Wi-Fi.

Las medidas se realizarán en dos canales distintos con distinta anchura: 20 MHz y 40 MHz, para poder observar que ventajas y desventajas presentaría las distintas anchuras de

canal. Para conseguir un canal de 40 MHz se utilizará la técnica de Channel Bonding que consistirá en la unión de dos canales de 20 MHz, siendo la utilización de esta técnica la principal novedad del trabajo. Además, al estar simulando tráfico TCP/Internet, siempre se enviará flujos de datos TCP.

Por último, se realizarán pruebas también teniendo en cuenta presencia de tráfico en la misma red Wi-Fi. Para ello, mientras el transmisor esté enviando información al receptor, los demás ordenadores de la sala estarán transmitiendo información. Con ello se observará el efecto que produce en un flujo de datos, que haya más dispositivos en la misma red Wi-Fi transmitiendo a la vez.

Una vez realizadas las medidas, éstas se capturarán utilizando el analizador de protocolos Wireshark, se filtrarán para obtener la información que se desee, se analizarán y se extraerán una serie de gráficas que mostrarán los resultados deseados.

1.2. Estructura de la memoria

El presente documento se estructura en 5 capítulos más dispuestos de la siguiente manera:

- En el capítulo 2 se explicarán una serie de aspectos teóricos relacionados con las redes Wi-Fi y las mejoras que aporta la misma 802.11n, además de una introducción al protocolo TCP.
- En el capítulo 3 se detallará todos los elementos que participarán en la realización de las medidas y los distintos tipos de escenarios que se han definido junto con todas las herramientas, programas, lenguajes de programación y modificaciones en los elementos que participarán en la realización de las medidas para que estas medidas se puedan realizar.
- En el capítulo 4 se explicará cómo se realizan las medidas, el filtrado y extracción de datos utilizando el programa Wireshark y un programa .c y finalmente la representación gráfica de esos datos usando un script Matlab.
- En el capítulo 5 se analizarán y representarán los resultados obtenidos para cada escenario seguidos de un comentario sobre estos.
- Por último, en el capítulo 6, se expondrá una conclusión general de todos los resultados y posibles líneas de trabajo futuro.

Capítulo 2. Aspectos Teóricos

Este apartado tratará de los aspectos técnicos relacionados con el estándar 802.11n, estándar sobre el cuál se va a trabajar y realizar las medidas. También se detallarán aspectos vitales que estarán presentes en las medidas como la agregación de tramas, y por último, el protocolo TCP, el funcionamiento de las ventanas de congestión en este protocolo, y los dos diferentes algoritmos de congestión que se van a utilizar en las medidas: New Reno y Cubic.

2.1. Conceptos básicos del estándar IEEE 802.11n

El estándar 802.11n fue publicado en octubre de 2009 concretamente. Este estándar es una versión extendida del estándar 802.11g por lo que será compatible con los dispositivos que empleen los estándares 802.11b y 802.11g con la adición de los dispositivos que utilicen el estándar 802.11a. El estándar 802.11n puede trabajar tanto en la banda de los 2,4 GHz como de los 5 GHz. La anchura de los canales puede ser tanto de 20 MHz como de 40 MHz.

La inclusión de canales de 40 MHz es una nueva ventaja incorporada en este estándar y se consigue con el empleo de la tecnología Channel Bonding mencionada en la Introducción, que consiste en la unión de dos canales adyacentes que no se solapan de 20 MHz, ya sean de la banda de frecuencias de los 5 GHz o de los 2,4 GHz, para la creación de un canal de anchura de 40 MHz, permitiendo incrementar la cantidad de datos transmitidos. Aunque esta tecnología es una novedad en este estándar, no está obligada su implementación, es una característica adicional que la puede incorporar o no el punto de acceso que use el estándar 802.11n. En este trabajo se empleará esta característica y se observará que mejoras presenta en las medidas.

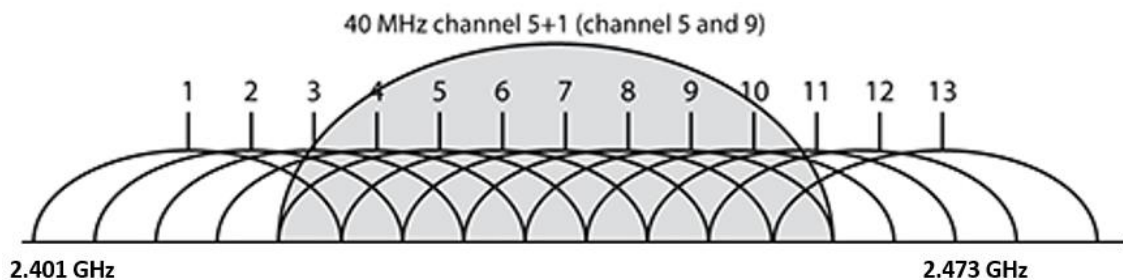


Figura 2.1. Channel bonding: representación canal de 40 MHz que aglutina los canales 5 y 9 y cuya anchura es la zona sombrada mientras que los demás semicírculos representan canales de 20 MHz

La principal novedad del estándar 802.11n es la técnica de transmisión MIMO, mencionada en la introducción. La tecnología MIMO consiste en transmitir y recibir flujos de información por múltiples antenas. Estos flujos de datos cada uno se denominan “spatial stream” y el punto de acceso deberá como mínimo soportar dos spatial streams, siendo el uso de 3 y 4 spatial streams opcional. Las principales ventajas de MIMO son la diversidad de antenas para la transmisión de datos y la diversidad espacial, que, aunque ya se podía emplear esta última característica en estándares anteriores, no se empleaba simultáneamente para conseguir un multiplexado espacial real como el que ofrece el estándar 802.11n. El beneficio de MIMO es

menor cuando hay una clara línea de vista, ya que las señales multiruta llegan más tarde que la señal de línea de visión, pero debido a la diversidad de señales multiruta que proporciona la tecnología MIMO permite al receptor recuperar los mensajes de la señal, lo cual hace esta tecnología muy apta para entornos interiores con obstáculos.

Otra ventaja que proporciona MIMO para la transmisión de información es SDM (Multiplexado de división espacial). Consiste en multiplexar espacialmente flujos de datos independientes simultáneamente. SDM puede mejorar el rendimiento de la transmisión conforme el número de spatial streams es incrementado. Cada spatial stream (flujo espacial) requiere una antena discreta tanto en el transmisor como en el receptor. El único problema de tener un gran número de spatial streams es que cada spatial stream requiere una cadena de radio frecuencia y un convertidor de analógico a digital para cada antena lo cual incrementa significativamente el coste.

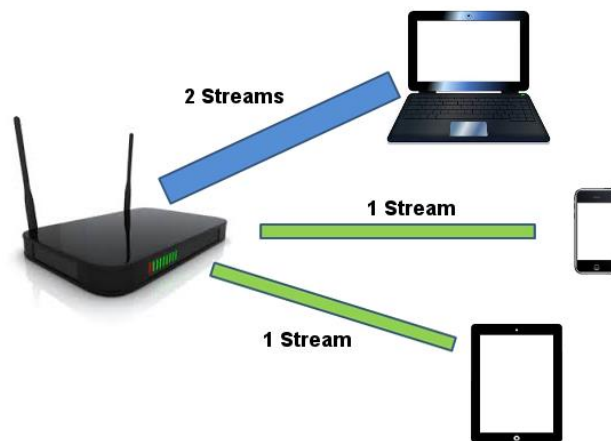


Figura 2.2. Dependiendo del número de antenas del punto de acceso y del dispositivo se decidirá el número de spatial streams

La velocidad de transmisión máxima en el estándar 802.11n es 600 Mbps, pero esto depende de que se den las características y condiciones adecuadas, como por ejemplo un canal de anchura de 40 MHz, un intervalo de guarda corto (400 ns frente a los 800 ns del intervalo de guarda normal), el número máximo de spatial streams, 4, el tipo de modulación digital usada (64-QAM para 600 Mbps) y el tipo de codificación empleada ($R=5/6$ para 600 Mbps). Variando cualquiera de estos valores la velocidad de transmisión cambiará, por lo que el estándar 802.11n proporciona un amplio rango de velocidades de transmisión. A continuación, se pondrá una lista de las velocidades de transmisión posibles según el número de spatial streams, anchura del canal, tipo de codificación, tipo de intervalo de guarda y la modulación digital empleada. Para destacar, todos los puntos de acceso que emplean el estándar 802.11n deberían tener una velocidad de transmisión máxima de 130 Mbps o superior.

Tabla 2.1. MCSs y velocidades de transmisión del estándar 802.11n

Index MCS	Modul.	Codif.	Spatial Streams	Canales 20 Mhz		Canales 40 Mhz	
				800 ns	400 ns	800 ns	400 ns
0	BPSK	1/2	1	6.5	7.2	13.5	15.0
1	QPSK	1/2	1	13.0	14.4	27.0	30.0
2	QPSK	3/4	1	19.5	21.7	40.5	45.0
3	16-QAM	1/2	1	26.0	28.9	54.0	60.0
4	16-QAM	3/4	1	39.0	43.3	81.0	90.0
5	64-QAM	2/3	1	52.0	57.8	108.0	120.0
6	64-QAM	3/4	1	58.5	65.0	121.5	135.0
7	64-QAM	5/6	1	65.0	72.2	135.0	150.0
8	BPSK	1/2	2	13.0	14.4	27.0	30.0
9	QPSK	1/2	2	26.0	28.9	54.0	60.0
10	QPSK	3/4	2	39.0	43.3	81.0	90.0
11	16-QAM	1/2	2	52.0	57.8	108.0	120.0
12	16-QAM	3/4	2	78.0	86.7	162.0	180.0
13	64-QAM	2/3	2	104.0	115.6	216.0	240.0
14	64-QAM	3/4	2	117.0	130.0	243.0	270.0
15	64-QAM	5/6	2	130.0	144.4	270.0	300.0
16	BPSK	1/2	3	19.5	21.7	40.5	45.0
17	QPSK	1/2	3	39.0	43.3	81.0	90.0
18	QPSK	3/4	3	58.5	65.0	121.5	135.0
19	16-QAM	1/2	3	78.0	86.7	162.0	180.0
20	16-QAM	3/4	3	117.0	130.0	243.0	270.0
21	64-QAM	2/3	3	156.0	173.3	324.0	360.0
22	64-QAM	3/4	3	175.5	195.0	364.5	405.0
23	64-QAM	5/6	3	195.0	216.7	405.0	450.0
24	BPSK	1/2	4	26.0	28.9	54.0	60.0
25	QPSK	1/2	4	52.0	57.8	108.0	120.0
26	QPSK	3/4	4	78.0	86.7	162.0	180.0
27	16-QAM	1/2	4	104.0	115.6	216.0	240.0
28	16-QAM	3/4	4	156.0	173.3	324.0	360.0
29	64-QAM	2/3	4	208.0	231.1	432.0	480.0
30	64-QAM	3/4	4	234.0	260.0	486.0	540.0
31	64-QAM	5/6	4	260.0	288.9	540.0	600.0

Otro beneficio del estándar 802.11n es su compatibilidad con los estándares 802.11g y 802.11a. Esta compatibilidad se conseguirá añadiendo un segundo preámbulo para MIMO. En canales de 40 MHz se repetirá este preámbulo en cada canal de 20 MHz que lo compone el canal de 40 MHz. Este método se denomina formato mixto. En el caso de que no haya dispositivos que empleen los antiguos estándares 802.11g y 802.11a, se podrá emplear el formato GreenField, que define un preámbulo exclusivo para 802.11n y mejora sus prestaciones. El formato Greenfield es una característica opcional y no hace falta que todos los dispositivos lo incorporen.

Debido a la compatibilidad con estándares anteriores, el estándar 802.11n seguirá empleando la tecnología de transmisión OFDM aumentando la codificación convolucional máxima a 5/6, añadiendo 4 subportadoras y una característica que es opcional será el empleo

de un intervalo de guarda más corto, concretamente de 400 ns, frente al intervalo de guarda anterior que era de 800 ns.

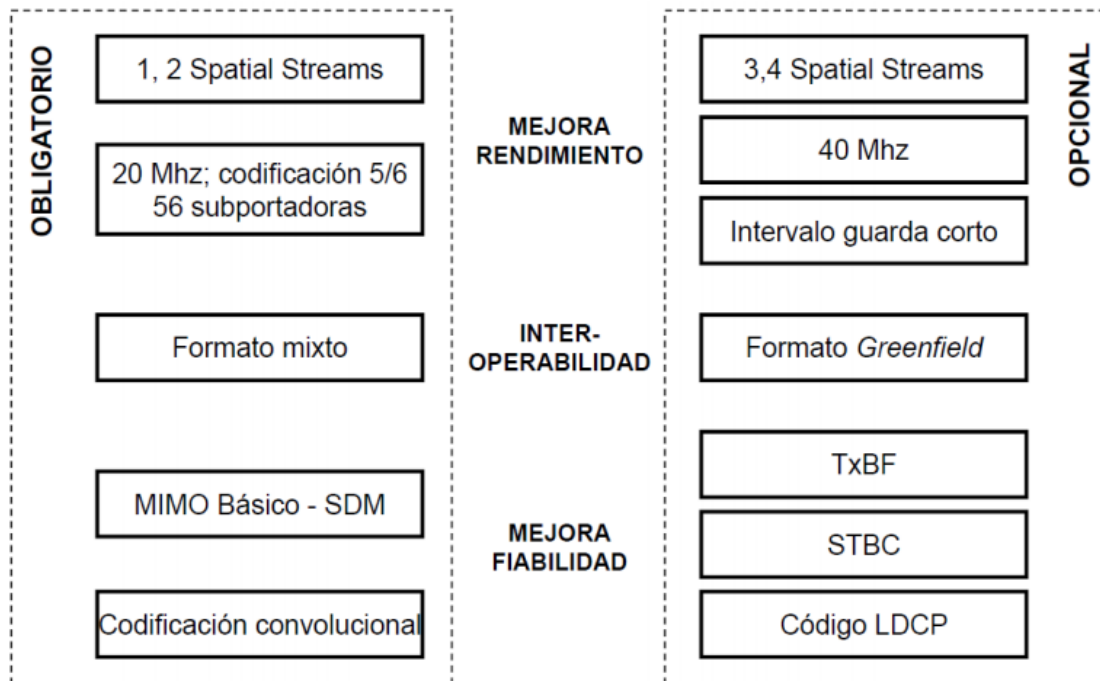


Figura 2.3. Características obligatorias frente a las características opcionales que puede presentar el estándar 802.11n, divididas según el beneficio que ofrecen: mejor rendimiento, mejor fiabilidad o inter-operabilidad

A continuación, se mencionará una serie de características opcionales que ofrece el estándar 802.11n para mejorar la fiabilidad. El primero de ellos es el Transmit Beamforming (TxBF) que básicamente es una característica que permite al punto de acceso concentrar efectivamente la señal en el lugar del usuario de la red Wi-Fi. Para ello es necesario conocer la situación del canal y actualmente es poco usado.

La segunda mejora opcional es el Space-Time Block Coding (STBC), que es una codificación que incorpora redundancia para mejorar la fiabilidad. A través de varios spatial streams se mandarían diferentes versiones codificadas de la señal.

La tercera y última mejora opcional es la codificación Low Density Parity Check (LDPC) que mejora las prestaciones del código convolucional tradicional. [1]

2.2. Agregación de tramas en el estándar 802.11n

La agregación de tramas es una característica diseñada en el estándar 802.11e en el que por primera vez se especifica la transmisión de múltiples tramas en lo que se conoce como Transmission Opportunity (TxOP). Posteriormente en el estándar 802.11n, esta técnica se especifica mediante los mecanismos de agregación conocidos como A-MPDU y A-MSDU, que se explicarán a continuación. Esto mejora drásticamente el rendimiento y eficiencia de las

transmisiones de datos debido al ahorro relacionado con la sobrecarga de múltiples transmisiones, utilizando una misma cabecera para un conjunto de mensajes. Para la confirmación de estas ráfagas de tramas en el caso de A-MPDU ya no se empleará la trama ACK sino la nueva trama BLOCK ACK, que confirma si se han recibido todas las tramas de una ráfaga correctamente o especifica si alguna no ha llegado correctamente. El ACK solo servía para confirmar tramas MPDU una por una mientras que el BLOCK ACK permitía la confirmación de una ráfaga de tramas MPDU, aumentando la eficiencia en la capa MAC. [1][2]

2.2.1. A-MSDU

A-MSDU consiste en la agrupación de MSDUs (MAC Service Data Unit), que son las unidades de servicio de datos que se recogen de la sub-capa LLC y que está sobre la sub-capa MAC. En este caso todas las tramas MSDU de una misma ráfaga A-MSDU comparten la cabecera MAC. El objetivo de A-MSDU es permitir múltiples MSDUs ser enviadas a un mismo receptor concatenadas como una MPDU (MAC Protocol Data Unit). Una MPDU consiste en el conjunto de datos recogidos de la sub-capa MAC. Para poder realizar una agregación A-MSDU es necesario la dirección destino (DA) y la dirección del emisor (SA) deben coincidir a las direcciones transmisor (TA) y receptor (RA) de la cabecera MAC, razón por la cual A-MSDU no puede ser empleado para tráfico Broadcast o Multicast. [1][2]

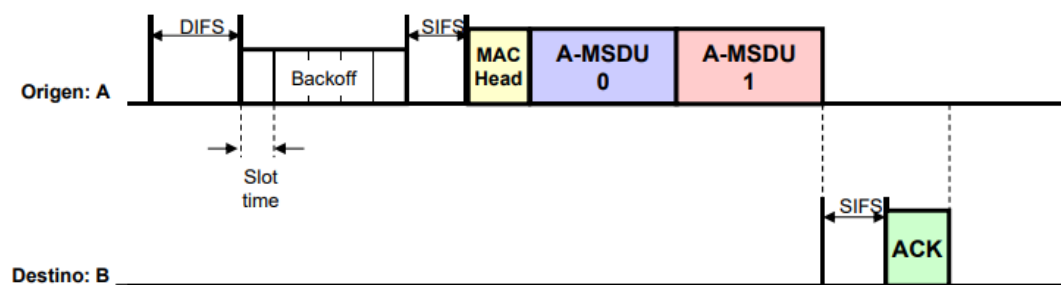


Figura 2.4. Funcionamiento agregación A-MSDU

2.2.2. A-MPDU

El objetivo de la A-MPDU es juntar varias tramas MAC, también denominadas como MPDUs (MAC Protocol Data Unit), en una transmisión en la que comparten la cabecera física, teniendo cada trama MAC su cabecera MAC no como en A-MSDU donde todas las tramas MSDU compartían una cabecera MAC. Aunque la ráfaga A-MPDU se transmita del mismo sitio, puede tener destinos diferentes cada trama MAC. Este método suele ser menos eficiente que el método A-MSDU, pero en entornos interiores con obstáculos como puede ser un entorno doméstico tiene una mayor eficiencia ya que gracias al BLOCK ACK permite la retransmisión selectiva de las tramas erróneas y no de toda la ráfaga A-MPDU, no como ocurre en A-MSDU. En este trabajo las transmisiones de datos se realizarán empleando A-MPDUs. El número de tramas MAC que contendrá una A-MPDU vendrá definido por los bytes que podrá transmitir como máximo en una A-MPDU y el tiempo máximo que debe durar una A-MPDU. En este caso nunca habrá limitación por exceso de tamaño, pero sí de tiempo. Se podrá comprobar que

dependiendo de la velocidad de transmisión de la A-MPDU contendrá más trama MAC (alta velocidad de transmisión) o no (baja velocidad de transmisión). [2]

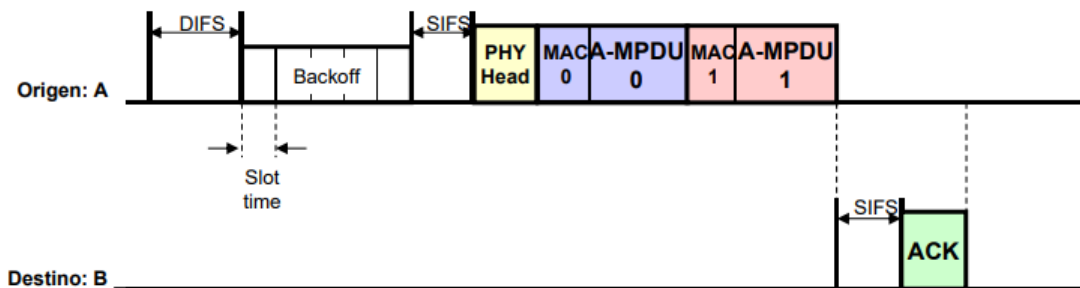


Figura 2.5. Funcionamiento agregación A-MPDU

2.3 Block ACK

El Block ACK, o reconocimiento en bloque en español, es una característica de la capa MAC de los estándares 802.11 que permite incrementar la tasa de transmisión efectiva, throughput, reduciendo la sobrecarga derivada del protocolo. El Block ACK fue inicialmente definido en el estándar 802.11e como una opción de la capa MAC para mejorar la eficiencia. El estándar 802.11n presentó una mejora de este mecanismo de Block ACK y estableció que fuese obligatorio que todos los dispositivos que soportasen el estándar 802.11n fueran capaces de utilizarlo. En vez de reconocer individualmente cada paquete recibido, una única trama Block ACK puede reconocer múltiples paquetes. Esto reduce el número de tramas ACK y sus correspondientes espacios entre tramas, produciendo un aumento en el throughput. Las tramas del bloque anterior que no han sido reconocidas serán transmitidas en el siguiente bloque. En la fase de conexión de un dispositivo a una red Wi-Fi, se negocian las distintas características relacionadas con aspectos como por ejemplo que tipo de Block ACK (Delayed o Immediate) se va a utilizar o el tamaño del buffer de tramas que se puede recibir sin esperar confirmación ACK. [3]

Una vez el transmisor envía múltiples tramas, acto seguido envía una trama de petición de Block ACK (Block ACK Request). El receptor responde con una trama Block ACK, reconociendo las tramas correctamente recibidas del bloque de transmisiones previo. Las tramas Block ACK Request y Block ACK son reconocidas si el modo Block ACK a destiempo (Delayed Block ACK) es utilizado, y no serán reconocidas si se utiliza la técnica Block ACK inmediato (Immediate Block ACK). El tipo de Block ACK usado en dispositivos que soportan el estándar 802.11n es el inmediato. Además, muchas veces las tramas Block ACK Request no se transmiten para evitar sobrecargas en las comunicaciones. Si el último MPDU tiene una política de ACK de control de QoS como ACK normal, en ese caso el receptor lo tratará como un Block ACK Request y enviará inmediatamente el Block ACK. Esto se denomina Block ACK implícito. [3][4]

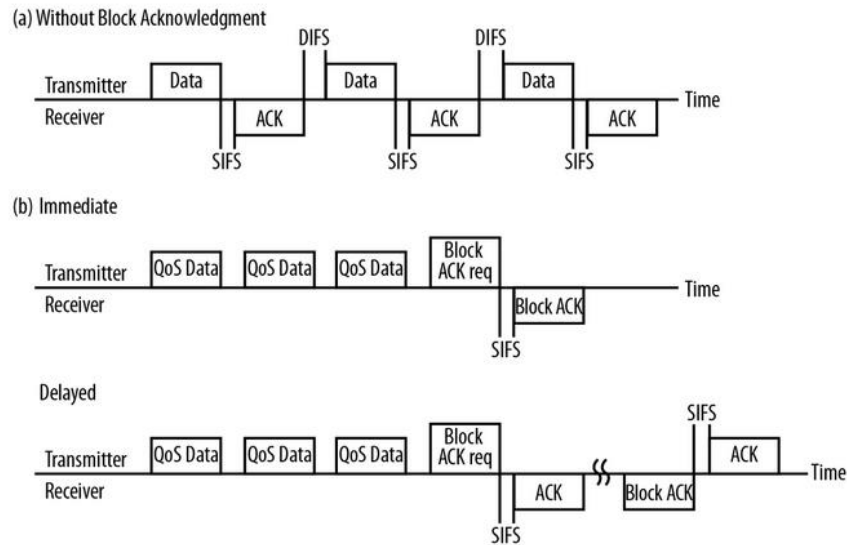


Figura 2.6. Reconocimiento simple vs Reconocimiento en Bloque [5]

2.4. Protocolo TCP

El protocolo TCP (Transmission Control Protocol) realiza las funciones de la capa de transporte de modelo OSI. Es orientado a la conexión y facilita el intercambio de mensajes entre distintos dispositivos en una red. Es el protocolo de Capa 4 más común en redes que utilizan IP (Internet Protocol). [6]

TCP se encarga de recoger los mensajes de aplicaciones/servidores y dividirlos en paquetes o “segmentos”, que serán transmitidos y redirigidos por los distintos dispositivos de una red al destino deseado. TCP numera cada paquete y los ordena de manera correcta previamente antes de entregar el mensaje al destino final. Como TCP es orientado a la conexión, TCP se asegura que la conexión se establece y mantiene hasta que el intercambio de información entre las aplicaciones y servidores se haya realizado correctamente. Por lo tanto, TCP es usado por aquellos protocolos de aplicación que necesiten que se entregue de manera garantizada un mensaje sin errores. Varios protocolos de aplicación utilizan TCP como por ejemplo HTTP o FTP. [6]

Para controlar el flujo de información y conseguir que TCP transmita el máximo de información posible de manera estable evitando congestionar la red, TCP utilizará complejos algoritmos para la gestión del número de segmentos que se podrán transmitir. Este proceso se denominará control de congestión TCP. Se tratará de limitar la tasa a la que se envía la información en función de la congestión que se percibe en la red. Si hay poca congestión, se incrementa la tasa de transmisión. Si aparece congestión, la tasa se reducirá. [7]

Para limitar la tasa de transmisión se utilizará la ventana de congestión, *cwnd*. El valor de la ventana de congestión determinará el número de bytes que el emisor podrá emitir sin esperar a recibir un ACK, una trama de reconocimiento. El número de bytes que podrá enviar el emisor siempre será el valor mínimo entre la ventana de congestión y el valor fijado por el buffer de recepción, siendo normalmente el valor limitante el de la ventana de congestión. [7]

$$\text{Bytes por confirmar} \leq \min \{cwnd, rwnd\}$$

El transmisor TCP detecta congestión cuando se pierden segmentos. Cuando se confirman segmentos la ventana de congestión aumentará y al detectar pérdidas la ventana de congestión se reducirá. Uno de los mayores problemas será establecer a qué velocidad crece la ventana de congestión. Para ello se utilizarán diferentes algoritmos de gestión como por ejemplo Tahoe o Reno. En este trabajo, se utilizarán dos algoritmos de control de congestión de red: Cubic y New Reno. [7]

2.4.1. Cubic

TCP Cubic es un algoritmo de control de congestión destinado para ser utilizado en enlaces con grandes anchos de banda, como son las conexiones actuales de Internet de la gran mayoría de las redes. En este tipo de enlaces si se emplea un algoritmo de control de congestión que incrementa lentamente la tasa de transmisión podría llevar a un desaprovechamiento de la capacidad de enlace, razón por la cual se ideó el algoritmo de congestión TCP Cubic dirigido a permitir un crecimiento exponencial de la ventana de congestión, pero sin llegar a sobrecargar la red. TCP Cubic se ha convertido en el segundo algoritmo de congestión más extendido, ya que es el que incorpora el kernel de Linux a partir de la versión 2.6.x. [6]

La expresión que permite calcular el tamaño de la ventana de congestión es la siguiente:

$$w = C \left(\Delta - \sqrt[3]{\beta \frac{W_{\max}}{C}} \right)^3 + W_{\max}$$

Donde W es el valor de la ventana de congestión, W_{\max} es el valor de la ventana de congestión al producirse la última pérdida, C y β son valores constantes para determinar la escalabilidad de la ventana de congestión y Δ es el tiempo transcurrido desde el último evento de pérdida. [6]

En la siguiente Figura 2.7, se puede observar la evolución de la ventana de congestión utilizando el algoritmo de control de congestión TCP Cubic. Cuando el valor de la ventana de congestión se sitúe alejado del valor de W_{\max} , el valor de la ventana de congestión crecerá exponencialmente. Una vez la ventana alcance ese valor, se mantendrá estable durante un tiempo en ese valor. Si no se llegase a producir un error durante un período de tiempo, se volvería a producir un incremento exponencial de la ventana de congestión hasta ocurrir un nuevo error. Una vez se produce un nuevo error, se fijará un valor de W_{\max} y se reducirá el valor de la ventana de congestión y se repetiría el proceso. [6]

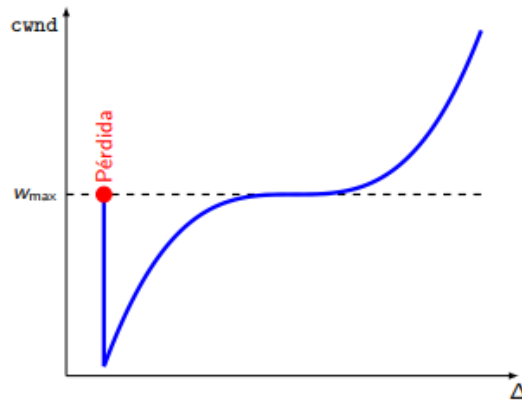


Figura 2.7. Evolución ventana de congestión en TCP Cubic

2.4.2. New Reno

New RENO propone una ligera modificación de la versión TCP-RENO, que le permite ser mucho más eficiente en caso de múltiples pérdidas de paquetes. [8]

Al igual que Reno, New-Reno también entra en retransmisión rápida (fast retransmission) cuando recibe múltiples paquetes duplicados, sin embargo se difiere de RENO en que no sale de la recuperación rápida (fast recovery) hasta que todos los datos que estaban pendientes en el momento de entrar en la recuperación rápida hayan sido reconocidos (ACK). De este modo supera el problema que tiene Reno de reducir el valor de CWND varias veces. [8]

La fase de transmisión rápida es la misma que en Reno. La diferencia se sitúa en la fase de recuperación rápida que permite múltiples retransmisiones en New Reno. Cada vez que new-Reno entra en fast recovery anota el segmento máximo que está pendiente. La fase de recuperación rápida procede como en Reno, sin embargo cuando se recibe un nuevo ACK, hay dos casos:

- Si se recibe un ACK de todos los segmentos que estaban pendientes cuando se entra en la recuperación rápida, entonces sale de la recuperación rápida y establece Cwnd a ssthresh, valor umbral que corresponde a la mitad de valor de la ventana de congestión al producirse una pérdida, y continua el control de congestión como si fuese Tahoe.
- Si el ACK es un ACK parcial entonces deduce que el siguiente segmento en línea se perdió y retransmite ese segmento y establece el número de ACKS duplicados recibidos a cero.

Sale de la recuperación rápida cuando todos los datos de la ventana son reconocidos.

[8]

Capítulo 3. Aspectos prácticos

En este capítulo se detallarán los diferentes escenarios sobre los cuales se van a realizar las medidas, que elementos participan y son necesarios para la realización de las medidas, la distancia entre estos elementos dependiendo del escenario, y por último, se comprobará la existencia de tráfico interferente de otras redes Wi-Fi que estén presentes en la realización de las medidas.

3.1. Descripción de la plataforma de medidas

En este apartado se detallarán las especificaciones de los diferentes elementos que estén presentes en la realización de las medidas. Para la transmisión y recepción de paquetes se utilizarán dos ordenadores idénticos. En el caso de que se desee generar tráfico interferente en la misma celda Ad-Hoc, se utilizarán otros 6 ordenadores que también estarán transmitiendo. Por lo tanto, en algunas medidas, habrá hasta 8 ordenadores trabajando a la vez. Estos ordenadores se conectarán a la red Wi-Fi utilizando adaptadores Wi-Fi Usb.

- Ordenadores

Como mínimo, en las medidas participaran 2 ordenadores, transmisor y receptor. Como máximo, habrá 8 ordenadores, transmisor, receptor y los otros 6 ordenadores generarán tráfico interferente. Los 8 ordenadores son idénticos:

- Placa base: Dell Inc. 0XJ5V0 Versión A03
- Procesador: Intel® Core™ i7-9700 CPU @ 3.00 GHz
- RAM instalada: 16 GB
- Versión Linux Ubuntu: Ubuntu 20.04.1 LTS

- Adaptador USB Wi-Fi

Para la realización de medidas se utilizará un adaptador Wi-Fi USB para conectarse a la red Wi-Fi ya sea tanto para la transmisión de paquetes, recepción de paquetes y monitorización del tráfico. El adaptador Wi-Fi USB utilizado TP-LINK modelo TL-WN821N (firmware v3) con un chipset Atheros AR9002U-2NG. Además, contendrá dos tipos de sistema en el chip (SOC = System on a Chip), el AR7010 y el AR9287, ambos modelos pertenecientes a Atheros. Este adaptador inalámbrico puede llegar a alcanzar los 300 Mbps de velocidad en una red Wi-Fi tipo 802.11n, que es en la cual se van a realizar las medidas, y solo trabaja en la banda de frecuencia de los 2,4 GHz.



Figura 3.1. Imagen del adaptador inalámbrico TP-LINK modelo TL-WN821N

Para ver la información de los dispositivos conectados a un puerto USB, como puede ser este adaptador USB Wi-Fi TL-WN821N, se abrirá una ventana de comandos y se introducirá el siguiente comando: “lsusb”.

```
rga91@Rodrigo:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 006: ID 0cf3:7015 Atheros Communications, Inc. TP-Link TL-WN821N v3 / TL-WN822N v2 802.11n [Atheros AR7010+AR9287]
Bus 001 Device 004: ID 062a:4106 Creative Labs
Bus 001 Device 003: ID 0e8f:0021 GreenAsia Inc. Multimedia Keyboard Controller
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 3.2. Respuesta al comando lsusb

La ventana de comandos devolverá información sobre los dispositivos USB conectados al ordenador, correspondiendo la línea 4 a la información relacionada al adaptador USB Wi-Fi, como su fabricante, Atheros Communications, el modelo del dispositivo empleado y los SOC (system on a chip) mencionados anteriormente.

Ahora se utilizará otro comando para conocer la información del driver de Atheros. Para ello se escribirá el siguiente comando en la línea de comandos: “dmesg | grep ‘ath’”

```
rga91@Rodrigo:~$ dmesg | grep 'ath'
[ 57.448675] usb 1-4: ath9k_htc: Firmware ath9k_htc/htc_7010-1.4.0.fw requested
[ 57.448722] usbcore: registered new interface driver ath9k_htc
[ 57.548912] usb 1-4: ath9k_htc: Transferred FW: ath9k_htc/htc_7010-1.4.0.fw, size: 72812
[ 57.618895] ath9k_htc 1-4:1.0: ath9k_htc: HTC initialized with 45 credits
[ 57.840380] ath9k_htc 1-4:1.0: ath9k_htc: FW Version: 1.4
[ 57.840381] ath9k_htc 1-4:1.0: FW RMW support: On
[ 57.840382] ath: EEPROM regdomain: 0x809c
[ 57.840382] ath: EEPROM indicates we should expect a country code
[ 57.840383] ath: doing EEPROM country->regdmn map search
[ 57.840383] ath: country maps to regdmn code: 0x52
[ 57.840384] ath: Country alpha2 being used: CN
[ 57.840384] ath: Regpair used: 0x52
[ 57.847879] ath9k_htc 1-4:1.0 wlx647002216fe2: renamed from wlan0
```

Figura 3.3. Respuesta al comando “dmesg | grep ‘ath’” donde ath representa el código de driver de Atheros

El resultado que ofrece la ventana de comandos muestra que el driver que se emplea recibe el nombre de ath9k_htc, que es un driver de código abierto, y la versión de este driver. Cabe destacar que en la última línea aparece como el driver ath9k_htc renombra el nombre del

interfaz Wi-Fi del ordenador “wlan 0” a “wlx647002216fe2”, donde los 12 dígitos añadidos corresponden a la dirección MAC del adaptador USB Wi-Fi.

3.2. Escenario

El escenario en cual se van a realizar las medidas es el laboratorio 128 en la primera planta de la Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Este laboratorio es una sala que dispone de 20 ordenadores para poder realizar las medidas y simular un posible tráfico interferente en la red Wi-Fi ad-hoc que se va a utilizar.

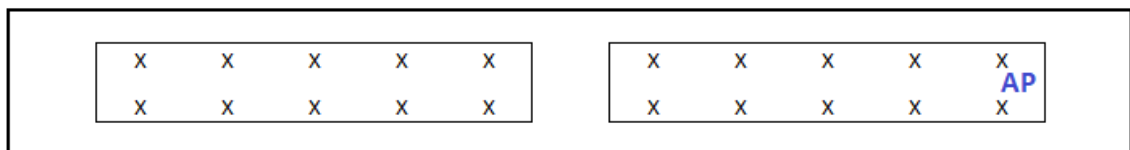


Figura 3.4. Distribución de los ordenadores del laboratorio 128

Como se puede observar en la figura, los 20 ordenadores, representados con una X, se encuentran distribuidos en dos mesas. Hay que tener en cuenta, que a pesar de haber 20 ordenadores, solo se utilizarán como máximo 8 ordenadores, debido a la falta de adaptadores USB Wi-Fi (sólo se dispone de 10 adaptadores Wi-Fi USB). En el transmisor y en el receptor se conectarán dos adaptadores Wi-Fi USB en cada uno. Un adaptador USB Wi-Fi trabajará en modo monitor capturando todos los paquetes Wi-Fi y beacons que se estén transmitiendo en el laboratorio por las redes Wi-Fi presentes. El segundo adaptador USB Wi-Fi servirá para crear/conectarse a la red Wi-Fi Ad-Hoc sobre la cual se van a realizar las medidas. Los 6 restantes adaptadores USB Wi-Fi se utilizarán en los 6 ordenadores restantes para que se puedan conectar a la red Ad-Hoc y crear ese tráfico interferente mencionado anteriormente. Por último, en azul indicado con las siglas AP “Access Point”, se encuentra el punto de acceso de la red “Tlmat”.

A pesar de que el entorno va a ser siempre el mismo, dependiendo de las distancias entre el receptor y el transmisor, se determinarán tres posibles casos de escenario.

3.2.1. Escenario de menor distancia

En este caso, el receptor y el transmisor estarán lo más cercano posible. La distancia entre transmisor y receptor es de 1 metro aproximadamente. El transmisor y receptor serán dos ordenadores que sean adyacentes. A continuación, se mostrará un plano indicando cuál es la posición del transmisor “TX” y del receptor “RX”, además de mostrar con una flecha roja la dirección del flujo de datos:

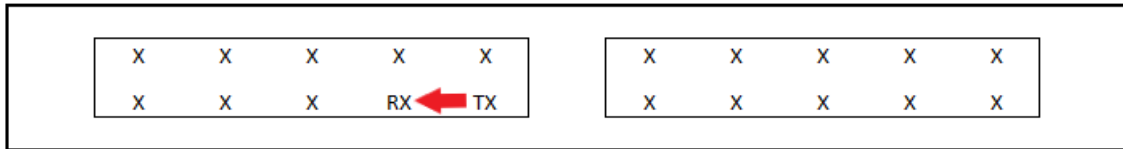


Figura 3.5. Plano aproximado para el escenario de menor distancia

La imagen real del escenario es la mostrada a continuación, indicando el receptor, usando el color amarillo, y el transmisor, usando el color rojo:

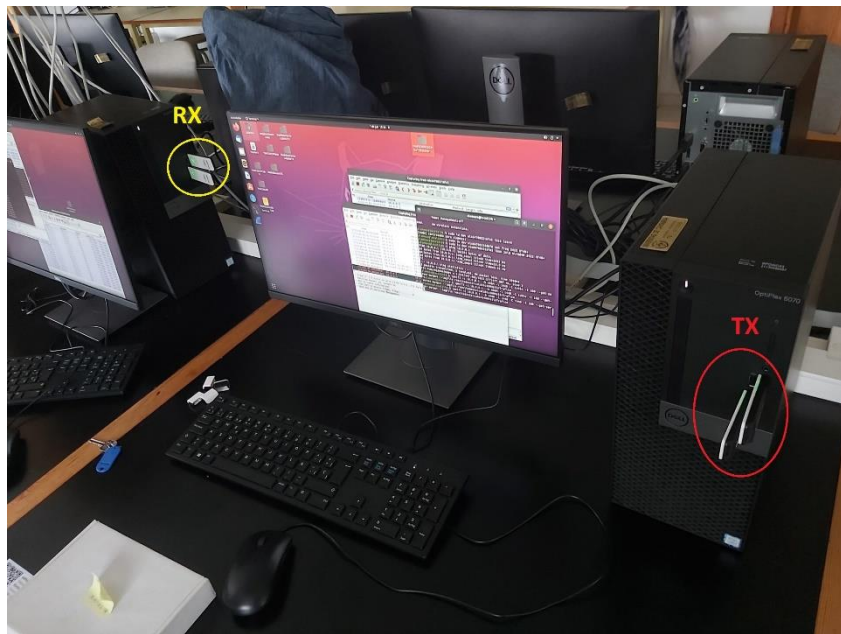


Figura 3.6. Escenario de menor distancia

Para el caso en que se realice las medidas con tráfico interferente, se añadirán a la red Wi-Fi Ad-hoc 6 ordenadores, los cuales generaran otros tres flujos de datos (1 flujo por cada par de ordenadores → TX y RX) que estarán constantemente transmitiendo durante las medidas. A continuación se mostrará el plano para este caso concreto, donde "IF" indicará que es un ordenador que genera o recibe tráfico interferente y se indicará también si es un receptor o un transmisor. Las flechas azules representan los flujos de datos de tráfico interferente, indicando las diferentes parejas Transmisor-Receptor que generan tráfico interferente:

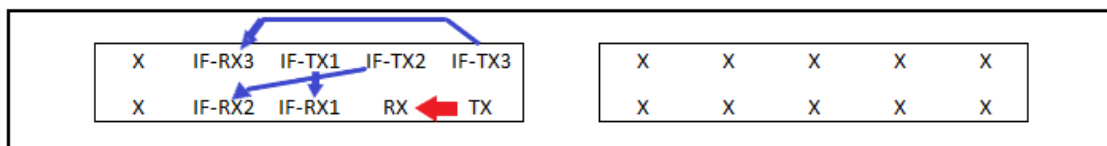


Figura 3.7. Plano para el escenario de menor distancia con tráfico interferente

3.2.2. Escenario de distancia media

El receptor y transmisor en este escenario se sitúan a una distancia media, concretamente de 5,30 metros aproximadamente. El ordenador receptor estará en un extremo de la sala y el ordenador transmisor en el centro de la sala. A continuación, se mostrará el plano con las posiciones del ordenador transmisor y receptor en la sala:

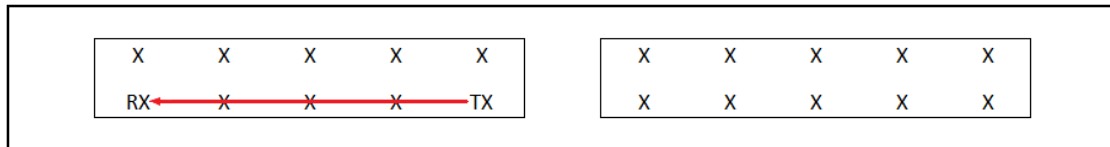


Figura 3.8. Plano aproximado para el escenario distancia media

La imagen real del escenario es la mostrada a continuación, indicando el receptor, usando el color amarillo, y el transmisor, usando el color rojo:



Figura 3.9. Escenario de distancia media

A continuación, se mostrará el plano para el caso en que se vaya a generar tráfico interferente durante las medidas. En el plano, al igual que antes en el escenario de menor distancia, se indicará que ordenadores son los encargados de generar tráfico interferente y la dirección de los flujos de datos:

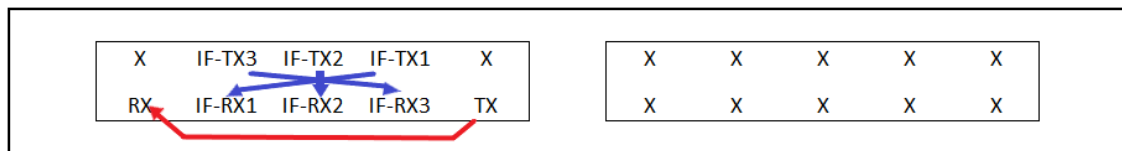


Figura 3.10. Plano para el escenario distancia de media con tráfico interferente

3.2.3. Escenario de mayor distancia

El receptor y transmisor en este escenario se sitúan a una distancia larga, concretamente de 11,60 metros aproximadamente. El ordenador receptor estará en un extremo de la sala y el ordenador transmisor en el otro extremo de la sala. A continuación, se mostrará el plano con las posiciones del ordenador transmisor y receptor en la sala:

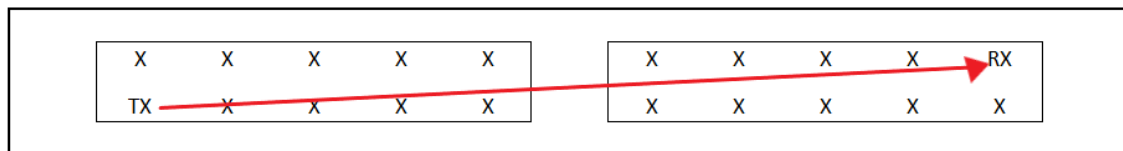


Figura 3.11. Plano aproximado para el escenario de mayor distancia

La imagen real del escenario es la mostrada a continuación, indicando el receptor, usando el color amarillo con una flecha ya que al situarse a una mayor distancia no se ve el ordenador receptor, y el transmisor, usando el color rojo:

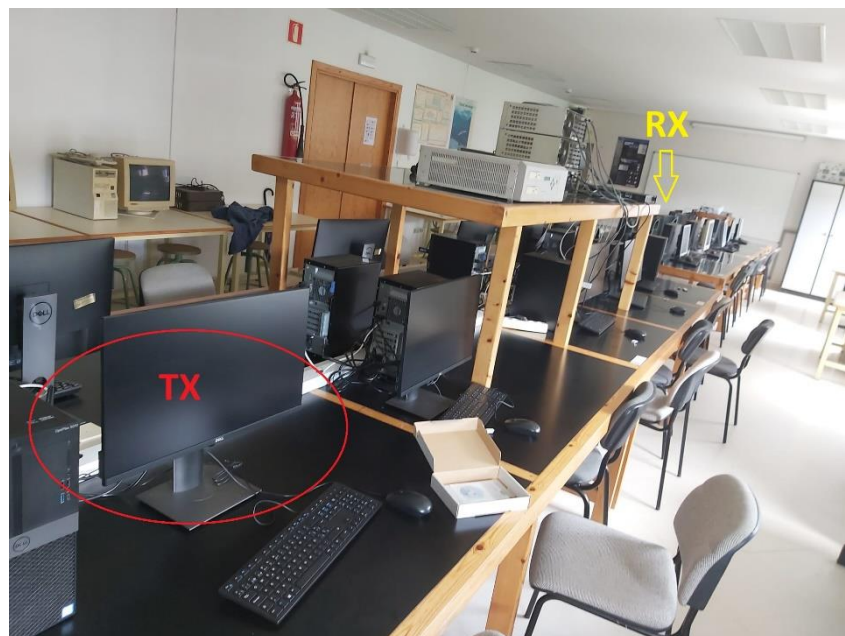


Figura 3.12. Escenario de mayor distancia

A continuación, se mostrará el plano para el caso en que se vaya a generar tráfico interferente durante las medidas. En el plano, al igual que antes en el escenario de menor distancia y distancia media, se indicará que ordenadores son los encargados de generar tráfico interferente y la dirección de los flujos de datos:

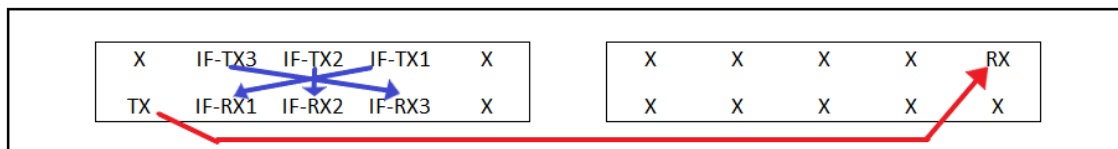


Figura 3.13. Plano para el escenario de mayor distancia con tráfico interferente

3.3. Tráfico interferente de otras redes

En este apartado se mostrará el distinto tráfico interferente de otras redes que podrían influir en la realización de las medidas en este trabajo. Para observar las diferentes redes Wi-Fi presentes en el entorno se ha utilizado el programa “inSSIDer”, que es una aplicación de escáner de redes de Wi-Fi, con la única desventaja que necesitas un sistema operativo Windows o OS X. Con esta aplicación se podrá observar los diferentes redes Wi-Fi presentes en el laboratorio. Se mostrará el tráfico interferente tanto para un canal de 20 MHz como para un canal de 40 MHz, cuya frecuencia central es 2.422 GHz para ambos canales, que son los que se van a utilizar en las medidas.

Comenzando con un canal de 20 MHz, cuyo nombre de la red Wi-Fi será “rodrigored”, la aplicación mostrará principalmente tres gráficas/tablas. La primera imagen mostrada a continuación mostrará una tabla con todas las redes Wi-Fi ordenadas según la potencia de señal recibida. La primera columna “SSID” indicará el nombre de la red Wi-Fi, la segunda columna “Signal” indicará la potencia de señal recibida de esa red Wi-Fi, la tercera columna indicará en que canal se sitúa la red Wi-Fi, la cuarta columna indicará el tipo de seguridad de la red Wi-Fi, la quinta columna indicará la dirección MAC de la red Wi-Fi, la sexta columna indica que estándar 802.11 utiliza esa red Wi-Fi y la última columna indica que topología se emplea en esa red Wi-Fi. En el caso de este trabajo, la red Wi-Fi “rodrigored”, por ejemplo, trabaja en el canal 3 (2422 MHz) y utiliza el estándar 802.11n. Por último, en la tabla se puede observar cómo hay una red Wi-Fi idéntica en todos los aspectos a la red de trabajo, compartiendo incluso nombre, “rodrigored”. Al estar utilizando una red Ad-Hoc, todos los integrantes de esta red Ad-Hoc emitirán beacons, por lo tanto la aplicación inSSIDer detectará a cada uno de los dispositivos integrantes de esta misma red Ad-Hoc como si formarían parte de distintas redes pero con las mismas características.





	SSID	SIGNAL ▼	CHANNEL	SECURITY	MAC ADDRESS	802.11	NETWORK TYPE
	rodrigored	 -46	3	Open	12:E2:66:5B:B9:E8	n	Ad Hoc
	rodrigored	 -46	3	Open	12:E2:66:5B:B9:E8	n	Ad Hoc
	LABTLMAT	 -57	1	WPA-Personal	00:14:BF:DE:77:8E	g	Infrastructure
	eduroam	 -67	6	WPA2-Enterprise	20:B3:99:FB:C0:39	n	Infrastructure
	UNICAN-i	 -67	6	Open	20:B3:99:FB:C0:38	n	Infrastructure
	UNICAN-i	 -73	6	Open	20:B3:99:A8:BA:79	n	Infrastructure
	eduroam	 -73	6	WPA2-Enterprise	20:B3:99:A8:BA:78	n	Infrastructure
	eduroam	 -82	11	WPA2-Enterprise	20:B3:99:A0:3C:B8	n	Infrastructure
	UNICAN-i	 -85	11	Open	20:B3:99:A0:3C:B9	n	Infrastructure
	MOVISTAR_33A7	 -91	11	WPA2-Personal	34:57:60:DE:33:A9	n	Infrastructure

Figura 3.14. Datos del entorno Wi-Fi para una red Wi-Fi de 20 MHz

En la siguiente Figura 3.15, se puede observar una gráfica que muestra la evolución de la potencia de señal recibida en amarillo para la red en la que se van a realizar las medidas y en rojo las redes que se superponen a esa red Wi-Fi de trabajo. Además, muestra datos de la red de trabajo mencionados anteriormente, con las novedades de que indica específicamente cuantas redes comparten canal con nuestra red de trabajo (“Co-Channel”), que en este caso es 0, ya que aunque indica que hay 1 red co-canal, debido al estar utilizando una red Ad-Hoc como se mencionó anteriormente, cada dispositivo perteneciente estará emitiendo beacons y la aplicación inSSIDer los detecta como redes Wi-Fi distintas, aunque son la misma red Wi-Fi realmente. El número de redes que se superponen a la red de trabajo, (“Overlapping”), que en este caso son 5. Por último, indica la máxima tasa binaria de la red de trabajo, “Max Rate”. En este caso indica que es 300 Mbps, pero es erróneo, ya que al ser un canal de 20 MHz con intervalo de guarda largo, lo máximo posible sería 130 Mbps. Este error es debido a que el adaptador USB Wi-Fi TPLink puede trabajar como máximo a una tasa binaria de 300 Mbps con un canal de 40 MHz. Por lo tanto el beacon de la red, indica que la velocidad binaria máxima es 300 Mbps, a pesar de que es un canal de 20 MHz.

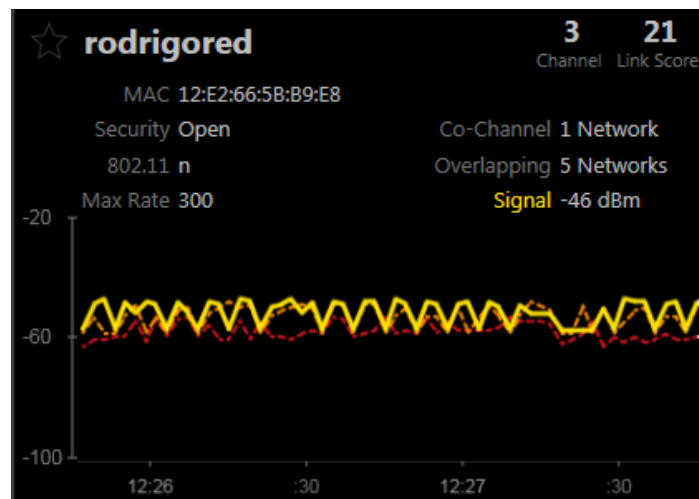


Figura 3.15. Representación potencia de señal recibida y otros datos de una red Wi-Fi de 20 MHz

Por último, en la figura 3.16, se podrá observar la representación gráfica del espectro de los 2.4 GHz. En esta figura se podrá ver en amarillo se representará la red Wi-Fi sobre la cual se van a realizar las medidas en el canal 3 y en gris el resto de las redes Wi-Fi del entorno.

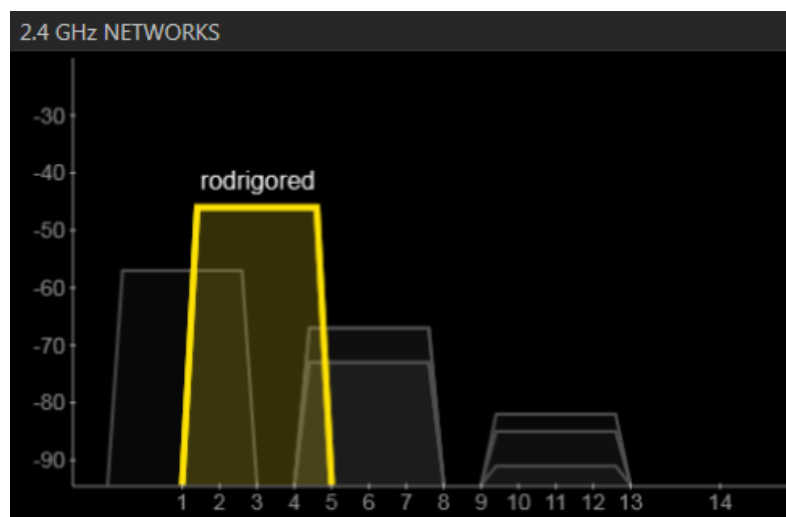


Figura 3.16. Representación gráfica del espectro de 2.4 GHz para un canal de 20 MHz

Utilizando la técnica Channel Bonding, se conseguirá un canal de 40 MHz. Al aumentar la anchura del canal se producirán varios cambios. En la siguiente Figura 3.17, aparecen una serie de datos sobre las redes Wi-Fi del entorno. En este caso, el SSID de la red Ad-Hoc creado es “mired”. Se puede observar que el canal de la red Wi-Fi de trabajo aparece un “3 + 7”. Esto es debido a que originalmente sin aplicar la técnica Channel Bonding, solo hay un canal de 20 MHz en la red Wi-Fi, en este caso el canal 3. Al utilizar Channel Bonding, para lograr un canal de 40 MHz se la sumado a ese canal 3, un canal superior en este caso el canal 7, para lograr un canal de 40 MHz que es la suma de dos canales de 20 MHz. Como dato, para lograr un canal de 40 MHz no es necesario unir tu canal original con un canal superior, también podría unirse con un canal inferior para lograr un canal de 40 MHz, pero para ello deberá haber suficiente espacio en el espectro de 2.4 GHz para lograr ese canal de 40 MHz y al ser el canal original el canal 3, no hay suficiente anchura espectral para lograr un canal de 40 MHz añadiendo un canal de 20 MHz de frecuencia inferior al canal 3.

Otros datos que aparecen en la tabla son la tasa binaria máxima, que es 300 Mbps en un canal de 40 MHz, y el tipo de red que se utiliza, el cuál es una red Ad-Hoc como ya se ha comentado anteriormente. El resto de los datos ya han sido comentados anteriormente y explicados en el entorno de una red Wi-Fi de 20 MHz. Al igual que antes en una red Wi-Fi de 20 MHz, la aplicación inSSIDer al estar utilizando una red Ad-Hoc detecta a cada uno de los dispositivos integrantes de esta misma red Ad-Hoc como si formarían parte de distintas redes pero con las mismas características.

	SSID	SIGNAL ▼	CHANNEL	SECURITY	MAC ADDRESS	MAX RATE	802.11	NETWORK TYPE
	LABTLMAT	-55	1	WEP	00:14:BF:DE:77:8E	54	g	Infrastructure
	mired	-56	3+7	Open	7A:86:22:29:04:43	300	n	Ad Hoc
	mired	-56	3+7	Open	7A:86:22:29:04:43	300	n	Ad Hoc
	eduroam	-65	6	WPA2-Enterprise	20:B3:99:FB:C0:39	130	n	Infrastructure
	UNICAN-i	-65	6	Open	20:B3:99:FB:C0:38	130	n	Infrastructure
	eduroam	-76	6	WPA2-Enterprise	20:B3:99:A8:BA:78	216	n	Infrastructure
	UNICAN-i	-77	6	Open	20:B3:99:A8:BA:79	216	n	Infrastructure
	eduroam	-78	11	WPA2-Enterprise	20:B3:99:A0:3C:B8	144	n	Infrastructure
	UNICAN-i	-80	11	Open	20:B3:99:A0:3C:B9	144	n	Infrastructure
	MOVISTAR_33A7	-91	6	WPA2-Personal	34:57:60:DE:33:A9	144	n	Infrastructure

Figura 3.17. Datos del entorno para una red Ad-Hoc de 40 MHz

En la siguiente Figura 3.18, se volverá a mostrar una gráfica con la potencia de señal recibida para una red Wi-Fi de 40 MHz. Los datos que aparecen en ella ya han sido explicados también anteriormente. Los datos más relevantes que aparecen en esta figura son el número de redes Wi-Fi que comparten canal (“Co-Channel”) con la red de trabajo es 0, ya que ocurre lo mismo que en una red Wi-Fi de 20 MHz, que detecta como una red Wi-Fi aparte al otro dispositivo perteneciente a la red Ad-Hoc, pero que realmente son la misma red Wi-Fi. El número de redes Wi-Fi que se superponen (“Overlapping”) a la red de trabajo es 11. Este incremento del número de redes que se superponen a la red de Wi-Fi de trabajo es obvio, debido a que al aumentar la anchura del canal Wi-Fi de 20 MHz a 40 MHz, entrará en conflicto con un mayor número de redes Wi-Fi del entorno al ocupar un mayor espectro de la banda de los 2.4 GHz.



Figura 3.18. Representación potencia de señal recibida y otros datos de una red Wi-Fi de 40 MHz

Por último, en la figura 3.19, se podrá observar la representación gráfica del espectro de los 2.4 GHz. En esta figura se podrá ver como aumenta considerablemente la anchura de un canal de 40 MHz, y a consecuencia de ello, entrar en conflicto con un mayor número de redes Wi-Fi del entorno.

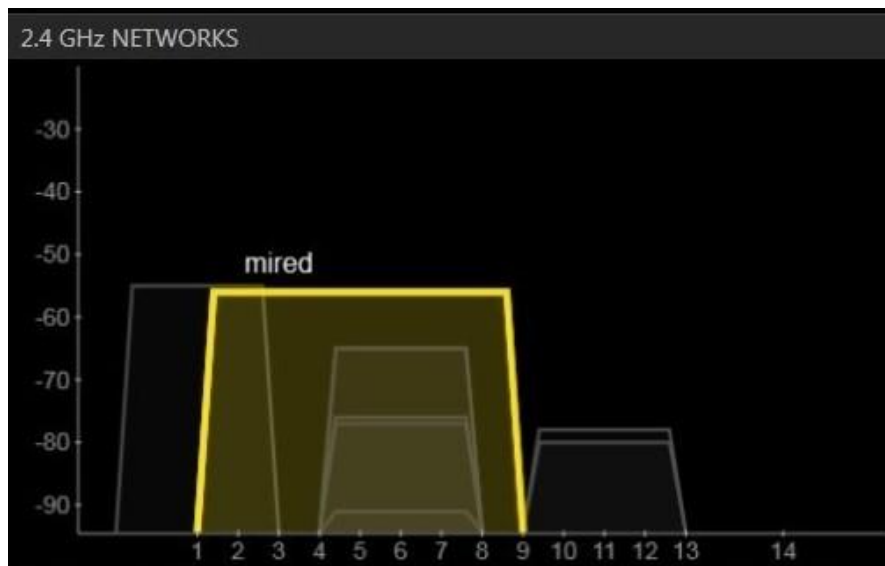


Figura 3.19. Representación gráfica del espectro de 2.4 GHz para un canal de 40 MHz

3.4. Herramientas empleadas en la campaña de medidas

Una vez definido el entorno y todos los agentes que entrarán en juego, se tendrá que instalar una serie de programas, aplicaciones y herramientas para poder realizar las medidas, procesarlas y analizarlas adecuadamente.

3.4.1. Herramienta de medida – Iperf3

Iperf3 es una herramienta que permite medir el rendimiento de una red. Tiene un funcionamiento modelo cliente-servidor, y puede generar flujos de datos para medir la velocidad de transmisión efectiva “throughput” entre dos puntos en una dirección o en ambas direcciones. Soporta varios parámetros relacionados con el tiempo, buffer, velocidad de transmisión deseada, protocolos (puede usar TCP y UDP, por ejemplo) o incluso el algoritmo de control de congestión que se desee utilizar con TCP. En cada prueba reporta el ancho de banda, pérdidas, throughput, y otros parámetros, como se verá más adelante.

3.4.2. Programas para análisis, obtención y representación de datos

Para pasos posteriores, como, por ejemplo, la representación de capturas, el filtrado de datos, la obtención de datos deseados o la creación de gráficas que representen esos datos, se tendrá que usar tres programas:

- El primero de ellos es el Wireshark. Se trata de un analizador de protocolos y con él se podrá monitorizar el flujo de datos que se produzca y analizarlo. También se podrá

utilizar algunas opciones de filtrado para que solo aparezcan las tramas deseadas, como, por ejemplo, sólo segmentos TCP.

- El segundo es Dev-C++. Este programa es un entorno de desarrollo integrado para programar en lenguaje C/C++. Utilizaremos Dev-C++ para crear el programa .C que se encargará de obtener los datos necesarios de la captura del flujo de datos.
- El último es Matlab. Este programa es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio, el lenguaje M. Con este programa escribiremos un programa .M para que una vez obtenidos los datos extraídos del programa .C, el programa .M se encargue de representarlos en forma de gráficas de la manera más adecuada.



Figura 3.20. Iconos de Wireshark, Dev-C++ y Matlab en este orden

Capítulo 4. Realización de las medidas

El objetivo de este capítulo será explicar cómo se realizan las medidas, las instrucciones introducidas en la ventana de comandos para crear la red Wi-Fi en modo Ad-Hoc en el cual se van a realizar las medidas, los programas que intervienen y sus funciones y objetivos dentro del proceso.

Se realizarán un total de 24 medidas. En cada medida utilizando la herramienta “iperf3” se generará un flujo TCP de una duración de 100 segundos, durante el cual el cliente enviará a la máxima velocidad posible, la mayor cantidad de información posible. Cada medida será diferente, dependiendo de los escenarios mencionados en el apartado 3, la anchura del canal Wi-Fi, 20 MHz o 40 MHz, el algoritmo de control de congestión TCP utilizado, y si hay o no presencia de tráfico interferente dentro de la misma red Wi-Fi.

Por cada medida, se realizarán 4 capturas de datos, 2 en el receptor y otras 2 en el transmisor. De esas 2 capturas, una de ellas corresponderá a una captura del interfaz 802.11n configurado en modo Ad-Hoc que está conectado a la red Wi-Fi creada, es decir, que sólo capturará el tráfico que vaya dirigido a ese ordenador en formato Ethernet II, mientras que la segunda captura corresponderá a una captura del interfaz en modo monitor, que básicamente captura todo el tráfico Wi-Fi que hay en ese punto.

Una vez definidos los objetivos, número de medidas a realizar y escenarios posibles, se procederá a realizar las medidas para extraer toda la información que sea útil utilizando un conjunto de códigos de programación creados por el autor del documento.

4.1. Creación de la red 802.11n

Para crear una celda en modo Ad-Hoc se utilizará la ventana de comandos. El primer paso será desactivar el gestor de redes Linux. Con ello se asegurará que el adaptador USB Wi-Fi se conecte en la red Wi-Fi deseada y que el gestor de redes no se conecte de manera automática a otra red. Siempre que se introduzca un comando se utilizará con privilegios de seguridad, es decir añadiendo al principio de cada línea de comando, “sudo”. El comando utilizado es el siguiente:

➤ `sudo service network-manager stop`

Una vez desactivado el gestor de redes de Linux, ya se podrá comenzar a trabajar con el adaptador USB Wi-Fi. El primer paso será conocer el nombre del interfaz del adaptador USB Wi-Fi. Para ello se utilizará el comando:

➤ `sudo iwconfig`

El resultado que aparece en pantalla es el siguiente en la Figura 5.1. En este trabajo, todos los nombres de los adaptadores USB Wi-Fi seguirán el mismo formato: wlx + una ristra de números y letras, que corresponde con la dirección MAC del adaptador USB Wi-Fi. Los datos que más interesan será el ESSID, que en este caso indica que no está conectado a ninguna red Wi-Fi, que no está asociado a ningún punto de acceso, que está trabajando en modo infraestructura (donde es necesaria la presencia de un router) y que utiliza el estándar 802.11, es decir, Wi-Fi.

```

alumnos@local210:~$ iwconfig
wlx64700221843a IEEE 802.11 ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:off

```

Figura 4.1. Resultados comando iwconfig

El siguiente paso será configurar el adaptador USB Wi-Fi en modo Ad-Hoc o IVSS (Independent Basic Service Set). Para ello primero habrá que desactivar temporalmente el interfaz del adaptador Wi-Fi USB, activar el modo Ad-Hoc del adaptador USB Wi-Fi, y volver a activar el interfaz. Los comandos utilizados son los siguientes:

- sudo ifconfig NombreInterfaz down
- sudo iw dev NombreInterfaz set type ibss
- sudo ip link set up dev NombreInterfaz

Si se Vuelve a introducir el comando iwconfig, el resultado es el siguiente que aparece en la Figura 5.2. Sigue indicando que no está conectado a ninguna red Wi-Fi, pero ya declara que está trabajando en modo Ad-Hoc.

```

alumnos@local210:~$ iwconfig
wlx64700221843a IEEE 802.11 ESSID:off/any
Mode:Ad-Hoc Frequency:2.412 GHz Cell: Not-Associated
Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:off

```

Figura 4.2. Adaptador USB Wi-Fi trabajando en modo Ad-Hoc

El siguiente paso será crear la red Wi-Fi. Para ello se introducirá la siguiente línea de comando:

- sudo iw dev NombreInterfaz ibss join mired 2422 HT20

Se deberá indicar el nombre de la red Wi-Fi o SSID, “mired”, la frecuencia central del canal Wi-Fi, en este caso el canal 3 o la frecuencia 2422 MHz, y por último, el ancho de banda de canal que se desea. Si no se indica que es “HT20” (High Throughput Canal de 20 MHz), la red Wi-Fi trabajará en un estándar inferior al 802.11n, el estándar 802.11g. Para crear una red Wi-Fi con una anchura de canal de 40 MHz, el último campo deberá ser HT40+ o HT40-. Como ya se explicó, para crear un canal de 40 MHz, se utiliza la técnica Channel Bonding. Básicamente, consiste en añadir un canal de 20 MHz al canal de 20 MHz ya existente. Esta adición se podrá realizar por el límite inferior o el límite superior del canal de 20 MHz existente. Como en este caso se trabajará en la frecuencia 2422 MHz, tendrá que ser añadido por el margen superior, debido a que no existen canales de 20 MHz con frecuencia central inferior a 2412 MHz (Canal 1) en el estándar 802.11. Si se vuelve a ejecutar el comando iwconfig, se observa que indica que ya está conectado a una red y la frecuencia central del canal:

```

alumnos@local210:~$ iwconfig
wlx64700221843a IEEE 802.11 ESSID:"mired"
Mode:Ad-Hoc Frequency:2.422 GHz Cell: Not-Associated
Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:off

```

Figura 4.3. Resultados de iwconfig después de crear una red Wi-Fi Ad-Hoc

Como se puede observar, aparece indicado que la red Wi-Fi no está asociado a ninguna celda. Esto es debido a que solo hay un ordenador en la red Wi-Fi en modo Ad-Hoc y es necesario un segundo ordenador para crear la celda. Para unir otro ordenador a la misma red Wi-Fi básicamente hay que seguir los pasos anteriores para crear una red Wi-Fi introduciendo los mismos parámetros.

El siguiente y último paso será añadir una dirección IP al dispositivo. Para ello se introducirá el siguiente comando:

- `sudo ip addr add DirecciónIP dev NombreInterfaz`

Para comprobar que la dirección IP se ha añadido correctamente se utilizarán dos métodos. El primero (Figura 5.4) corresponde a introducir el comando “`sudo ip addr show dev NombreInterfaz`” que mostrará las direcciones IP asignadas del dispositivo. La segunda consiste en básicamente realizar un ping a un ordenador de la red Wi-Fi y ver si funciona correctamente.

```

alumnos@local210:~$ sudo ip addr add 10.0.0.2/8 dev wlx64700221843a
alumnos@local210:~$ sudo ip addr show dev wlx64700221843a
3: wlx64700221843a: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 64:70:02:21:84:3a brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/8 scope global wlx64700221843a
        valid_lft forever preferred_lft forever
    inet6 fe80::6670:2ff:fe21:843a/64 scope link
        valid_lft forever preferred_lft forever

```

Figura 4.4. Comprobación Dirección IP

```

alumnos@local210:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=26.8 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.68 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=1.60 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.55 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=1.57 ms
^C
--- 10.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 1.551/6.629/26.751/10.061 ms

```

Figura 4.5. Resultado Ping

Introduciendo el siguiente comando, se podrá conocer información del dispositivo y de la red Wi-Fi en modo Ad-Hoc creada reciente:

- `sudo iw dev NombreInterfaz info`

```

alumnos@local210:~$ sudo iw dev wlx64700221843a info
Interface wlx64700221843a
    ifindex 3
    wdev 0x1
    addr 64:70:02:21:84:3a
    ssid mired
    type IBSS
    wiphy 0
    channel 3 (2422 MHz), width: 20 MHz, center1: 2422 MHz
    txpower 20.00 dBm

```

Figura 4.6. Información del dispositivo y la red Wi-Fi a la que esté conectada

Por último, en el caso de que se desee crear una nueva red Wi-Fi, sin cambiar ninguna dirección IP, con una anchura de canal de 40 MHz, simplemente hay que introducir los siguientes comandos. Primero habrá que salirse de la actual red Wi-Fi y acto seguido crear o unirse a una red Wi-Fi nueva:

- `sudo iw dev NombreInterfaz ibss leave`
- `sudo iw dev NombreInterfaz ibss join mired 2422 HT40+`

4.2. Captura de datos

Con los ordenadores completamente preparados y todos los programas necesarios instalados, ya se podrá comenzar a realizar las pruebas. Se desconectará los ordenadores de cualquier red Ethernet o Wi-Fi a la que puedan estar conectados. Al estar ya creada la red Wi-Fi en modo Ad-Hoc en la cual se van a realizar medidas, ya habrá un adaptador USB Wi-Fi listo para capturar datos de la red Wi-Fi Ad-Hoc. Solo quedará colocar un segundo adaptador USB Wi-Fi en modo monitor en el receptor y en el transmisor para capturar todo el tráfico Wi-Fi presente en el entorno sobre el canal seleccionado.

- Modo monitor

Para poner una adaptador USB Wi-Fi en modo monitor se seguirá un proceso similar al de crear o unirse a una red Wi-Fi en modo Ad-Hoc. Primero se deberá desactivar temporalmente la interfaz del dispositivo, se deberá poner el dispositivo en modo monitor, se volverá a activar la interfaz del dispositivo, y por último, se introducirá un comando para determinar en qué frecuencia deberá estar capturando paquete y para qué tipo de red, HT20 o HT40+, en este trabajo. Los comandos utilizados son los siguientes:

- `sudo ifconfig NombreInterfaz down`
- `sudo iw dev NombreInterfaz set type monitor`
- `sudo ip link set up dev NombreInterfaz`
- `sudo iw dev NombreInterfaz set freq 2422 HT20`

Para comprobar que se han realizado correctamente los cambios, se introducirá comandos como “iwconfig” (Figura 5.7) o “sudo iw dev NombreInterfaz info” (Figura 5.8), mencionados anteriormente

```
wlx647002217c67 IEEE 802.11 Mode:Monitor Frequency:2.422 GHz Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:off
```

Figura 4.7. Resultado comando iwconfig para un dispositivo en modo monitor

```
alumnos@local210:~$ iw dev wlx647002217c67 info
Interface wlx647002217c67
    ifindex 4
    wdev 0x100000001
    addr 64:70:02:21:7c:67
    type monitor
    wiphy 1
    channel 3 (2422 MHz), width: 20 MHz, center1: 2422 MHz
    txpower 20.00 dBm
```

Figura 4.8. Información de un dispositivo en modo monitor

- Wireshark

Una vez los ordenadores estén conectados a la red Wi-Fi modo Ad-Hoc creada y haya un segundo adaptador USB Wi-Fi en modo monitor, se podrá comenzar a realizar medidas. Pero antes de ello, se deberá capturar todo el flujo de datos de la medida. Para ello previamente hay que utilizar Wireshark. Nada más abrir Wireshark se pedirá en que interfaz se desea capturar. Como ya se mencionó en la introducción de este apartado, se va a capturar tanto en el interfaz monitor como en el interfaz Ad-Hoc conectado a la red Wi-Fi creada, ya sea en el transmisor o en el receptor. Por lo tanto, se deberá abrir dos veces Wireshark en cada ordenador para capturar en dos interfaces diferentes, una interfaz por ventana de Wireshark. Una vez se hace click en el interfaz deseado, Wireshark ya comienza a capturar, pudiendo ya capturar el flujo de datos que va a generar la herramienta de medida iperf3.

- Iperf3

Con Wireshark en pleno funcionamiento, ya se podrá comenzar a realizar las medidas utilizando la herramienta de medida iperf3. Iperf3 tiene una serie de comandos y características que permiten definir como se desea realizar las medidas, por ejemplo, indicando quién es el transmisor, la duración de la medida o el algoritmo de control de congestión TCP utilizado.

- Receptor

En el ordenador receptor se abrirá un terminal y se escribirá la siguiente línea de comando de la cual se explicarán sus parámetros, para que el ordenador receptor esté listo para recibir la transmisión de datos:

iperf3 -s

-s = indica que este ordenador es el servidor y por tanto actúa como el receptor de datos

- Transmisor

En el ordenador transmisor, una vez el ordenador receptor esté listo para recibir la transmisión de paquetes, se abrirá una terminal y se introducirá la siguiente línea de comando, explicando a continuación que significan todos los parámetros que contiene:

```
iperf3 -c "dirección IP destino" -b 500M -l 1448 -C "algoritmo de control de congestión"  
-t 100 --get-server-output --logfile "Nombre fichero resultados"
```

-c "Dirección IP" = indica que este ordenador es el transmisor de datos, el cliente. A continuación, se deberá indicar la dirección IP del servidor al que se va a transmitir información.

-b "Velocidad de transmisión" = indica la velocidad de transmisión objetivo. En este caso, se ha indicado que es 500 Mbps. Debido a que es una tasa de transmisión muy alta, el transmisor siempre intentará enviar a la máxima velocidad posible (130 Mbps en un canal de 20 MHz o 300 Mbps en un canal de 40 MHz), ya que nunca alcanzará esos 500 Mbps. Otra forma de indicar que envíe a la máxima velocidad posible es introduciendo en este campo un 0, ya que iperf3 lo interpretará que se deberá transmitir a la máxima velocidad posible.

-l "Número" = indica la longitud del buffer. Para hacernos una idea, esto indica el tamaño de los datos enviados en Bytes sin contar las cabeceras. En este caso se enviarán segmentos de datos de 1448 Bytes.

-C "algoritmo de control de congestión" = indica el algoritmo de control de congestión TCP que se va a emplear en la medida. En este trabajo son dos: New Reno y Cubic.

-t = indica el tiempo de duración en segundos de la medida. En este trabajo todas las medidas serán de 100 segundos. La razón de realizar medidas de 100 segundos es debido a que si se realizan medidas cortas, no da tiempo para que la ventana de congestión evolucione adecuadamente, y si se utilizan medidas de más de 100 segundos, el peso de los archivos obtenidos en Wireshark crece exponencialmente (En una medida de 100 segundos en algunos casos hay archivos que alcanzan casi el tamaño de 2 GB)

--get-server-output = indica que el cliente transmisor reciba también los resultados mostrados por pantalla del receptor una vez finalice la medida.

--logfile "Nombre fichero resultados" = indica que los resultados de la medida se escriban en un fichero.

Una vez ejecutada esta línea de comando, se producirá la transmisión de paquetes, y una vez acabada aparecerá en el ordenador transmisor los resultados sacados de la prueba realizada.

- Resultados

A continuación, se muestran los resultados de una medida. Al ser medidas grandes, se separará en tres imágenes todo lo necesario para comprender los resultados que devuelve iperf3:

```
Connecting to host 10.0.0.1, port 5201
[ 6] local 10.0.0.2 port 53130 connected to 10.0.0.1 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 6]  0.00-1.00    sec  10.0 MBytes  84.1 Mbits/sec    0   276 KBytes
[ 6]  1.00-2.00    sec   9.65 MBytes  80.9 Mbits/sec    0   303 KBytes
[ 6]  2.00-3.00    sec   9.88 MBytes  82.8 Mbits/sec    0   303 KBytes
[ 6]  3.00-4.00    sec   9.79 MBytes  82.2 Mbits/sec    0   303 KBytes
[ 6]  4.00-5.00    sec   9.74 MBytes  81.7 Mbits/sec    0   303 KBytes
[ 6]  5.00-6.00    sec   9.89 MBytes  83.0 Mbits/sec    0   303 KBytes
[ 6]  6.00-7.00    sec   9.81 MBytes  82.3 Mbits/sec    0   303 KBytes
[ 6]  7.00-8.00    sec   9.90 MBytes  83.0 Mbits/sec    0   303 KBytes
[ 6]  8.00-9.00    sec   9.70 MBytes  81.4 Mbits/sec    0   338 KBytes
[ 6]  9.00-10.00   sec   9.86 MBytes  82.7 Mbits/sec    0   338 KBytes
```

Figura 4.9. Resultado iperf3 (1)

Se puede observar en la imagen como lo primero que hace el transmisor cliente es conectarse con el servidor, indicando además que puertos están utilizando el transmisor y el receptor en la medida. A continuación comenzará la medida y el transmisor comenzará a enviar datos al servidor. Como aparece en la imagen, iperf3 muestra los resultados por intervalos de tiempo de 1 segundo. En cada intervalo de tiempo, en la segunda columna, indicará la cantidad de datos transmitidos en ese intervalo (Transfer). También mostrará tasa de transmisión efectiva, también conocido como throughput, en la tercera columna (Bitrate), que básicamente es la velocidad de transmisión real. En la cuarta columna (Retr), indicará si se ha producido alguna retransmisión TCP, y en la última columna, se mostrará la evolución de la ventana de congestión.

En la siguiente Figura 4.10, se muestra el final del fichero de resultados de una medida utilizando iperf3. Una vez finaliza la medida, se muestra un resumen con información generalizada de la medida que son la duración de la medida (100 segundos) en la primera columna, la cantidad de datos que se han transmitido (Transfer) en la segunda columna, la tasa de transmisión efectiva o throughput en la tercera columna (Bitrate) y el número de retransmisiones TCP que se han producido durante la medida en la última columna (Retr). Como se puede observar, hay dos filas de resultados. La primera fila corresponde a los resultados reportados por el transmisor, mientras que la segunda fila corresponde a los resultados reportados por el receptor. Los resultados reportados tanto por el transmisor como por el receptor son casi idénticos. La razón por la que varían ligeramente, siendo siempre los valores obtenidos del transmisor superiores a los resultados del receptor, es porque mientras el servidor está cerrando la conexión, hay un brevísimo instante durante el cual, el transmisor todavía sigue transmitiendo.

```
[ 6] 98.00-99.00 sec  9.50 MBytes  79.7 Mbits/sec    0   303 KBytes
[ 6] 99.00-100.00 sec  9.38 MBytes  78.6 Mbits/sec    0   303 KBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 6]  0.00-100.00 sec  852 MBytes  71.5 Mbits/sec    5
[ 6]  0.00-100.01 sec  850 MBytes  71.3 Mbits/sec    0
```

sender
receiver

Figura 4.10. Resultados finales iperf3 (2)

Por último, si a la hora de realizar la medida, en el transmisor, al introducir la línea de comando, se ha añadido la característica “--get-server-output” que ya se ha mencionado anteriormente, también se mostrará los resultados aportados por el servidor. Como se puede apreciar, no aportan ningún tipo de información desconocida hasta el momento, aportando incluso menos información que el transmisor. Con esto se podrá ver la evolución del throughput desde el punto de vista del servidor. Los resultados del servidor se encontrarán al final del fichero de resultados de iperf3.

Server output:

Server listening on 5201

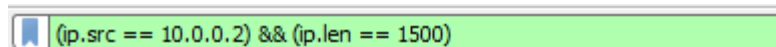
```
Accepted connection from 10.0.0.2, port 53128
[ 5] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 53130
[ ID] Interval          Transfer      Bitrate
[ 5]  0.00-1.00      sec  9.00 MBytes  75.5 Mbits/sec
[ 5]  1.00-2.00      sec  9.56 MBytes  80.2 Mbits/sec
[ 5]  2.00-3.00      sec  9.87 MBytes  82.8 Mbits/sec
```

Figura 4.11. Resultados finales iperf3 del servidor (3)

- Filtrado

Una vez capturada la transmisión de paquetes, se tendrán que filtrar los paquetes que se necesiten y pasarlo a un formato de fácil manejo, como puede ser .txt. Para ello se utilizará la función de filtrado de Wireshark. Sólo se tendrá que introducir los filtros en el buscador de la parte superior. Se tendrá que filtrar dos veces. El primer caso será para obtener la potencia de señal recibida del receptor y la tasa de transmisión, que se obtiene de la captura del modo monitor en el receptor, y el segundo caso que será para obtener el tiempo que se otorga al transmisor para poder enviar datos, que se encuentra en la captura modo monitor del receptor.

Para el primer caso hay que realizar el siguiente filtrado, para conseguir solamente los paquetes TCP transmitidos:



(ip.src == 10.0.0.2) && (ip.len == 1500)

Figura 4.12. Filtros empleados para obtener los segmentos TCP

En la imagen superior se puede apreciar los filtros empleados que son:

- Ip.src == 10.0.0.2 → filtra datagramas con esa dirección IP de origen.
- Ip.len == 1500 → filtra tráfico cuyo tamaño del datagrama IP sea de 1500 Bytes.

Antes de nada, una vez se tengan representados los paquetes que se deseen, se deberá elegir que datos se van a necesitar en las medidas y abrir las ventanas donde se encuentran. Por ejemplo, en este caso se querrá saber la velocidad de transmisión, potencia de señal,

identificador IP de los paquetes y el tamaño del paquete. Para ello se tendrá que desplegar las ventanas “802.11 radio information” y “Internet Protocol Version 4”:

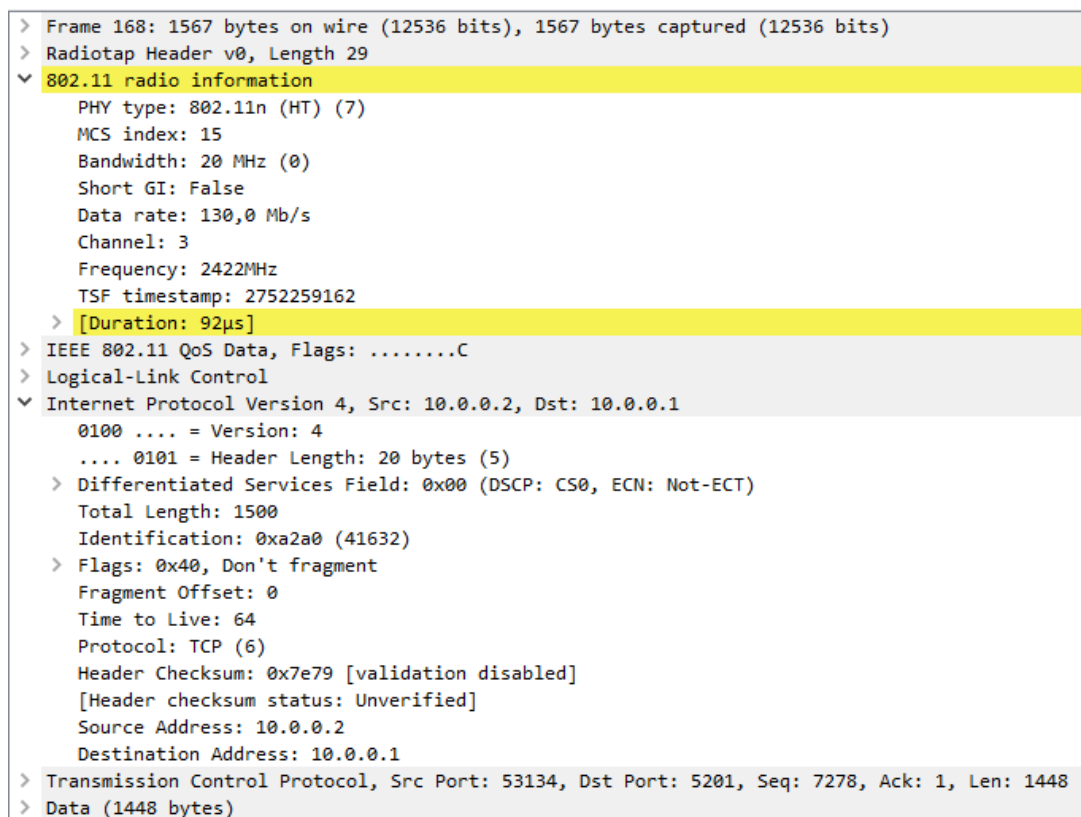


Figura 4.13. Imagen con las ventanas descritas desplegadas de un paquete TCP.

Para el segundo caso, el de obtener los tiempos que se le otorgan al transmisor para poder transmitir información, habrá que revisar las tramas CTS “Clear To Send”. Los filtros utilizados son los siguientes:

`(wlan.ra == 64:70:02:21:6f:e2) && (wlan.fc.type_subtype == 0x001c)`

Figura 4.14. Filtros empleados para obtener tramas CTS

En la imagen superior se puede apreciar los filtros empleados que son:

- wlan.ra == 10.0.0.2 → filtra datagramas con esa dirección física como destino, que corresponde al transmisor.
- Wlan.fc.type_subtype == 0x001c → filtra todas las tramas CTS del archivo.

Al igual que antes, se deberá elegir que datos se van a necesitar en las medidas y abrir las ventanas donde se encuentran. Por ejemplo, en este caso se querrá saber el tiempo que se otorga al transmisor para enviar información. Para ello se tendrá que desplegar la ventana “IEEE 802.11 Clear-to-send”. En la imagen inferior, por ejemplo, indica que el transmisor tiene 1152 microsegundos de tiempo para transmitir.

```

> Frame 238: 52 bytes on wire (416 bits), 52 bytes captured (416 bits)
> Radiotap Header v0, Length 38
> 802.11 radio information
▼ IEEE 802.11 Clear-to-send, Flags: .....C
  Type/Subtype: Clear-to-send (0x001c)
  > Frame Control Field: 0xc400
    .000 0100 1000 0000 = Duration: 1152 microseconds
    Receiver address: Tp-LinkT_21:6f:e2 (64:70:02:21:6f:e2)
    Frame check sequence: 0x60fada87 [unverified]
    [FCS Status: Unverified]

```

Figura 4.15. Imagen con las ventanas descritas desplegadas de una trama CTS

Una vez hecho esto, tanto para el primer caso como para el segundo caso, se podrá ya pasar los datos a un archivo .txt. Para ello hay que hacer click en File → Export Packet Dissections → As plain text. Se abrirá una ventana y se comprobará que este seleccionada la opción de mostrar cómo se ha elegido, con los filtros elegidos y las ventanas abiertas elegidas. En mi caso esta seleccionado la opción “Packet Summary line” que añade una línea antes de cada paquete indicando una serie de características sobre este y que es la opción por defecto. Por último, se guardará el archivo en el directorio que se desee y ya estaría listo el archivo .txt de la captura.

Cabe destacar que la representación de Wireshark de las tramas obtenidas es una aproximación, al ser una trama A-MPDU todas las tramas de un grupo pertenecen a ella compartiendo la misma cabecera física. En Wireshark aparecerán como tramas independientes (aunque bien es cierto que hay un campo que te indica que forma parte de una A-MPDU) con la misma cabecera física siendo la última trama MPDU de esta ráfaga la que contenga información de la potencia de señal recibida de la A-MPDU y todas ellas con una cabecera MAC que las diferencia. (Ver apartado 2.2.2 A-MPDU)

4.3. Obtención datos con C

Una vez se tengan las capturas en formato texto representada de la manera deseada, se deberá extraer únicamente los datos con los que se va a trabajar. Para ello se creará una serie de ficheros que contengan esos datos y luego esos ficheros se pasarán a Matlab para realizar la representación gráfica.

Primero de todo, hay que aclarar que para la obtención de datos y creación de ficheros se va a utilizar dos programas .C creados con Dev-Cpp. Se ha optado por el lenguaje de programación C por ser un lenguaje sencillo y sobre el que ya se tiene conocimientos sobre él.

El nombre de los programas .C son “ProcesaTCP.c” y “ProcesaCTS.c” y los podréis encontrar en el anexo.

Los programas siguen una estructura similar dividida en 3 partes: inicialización, lectura y obtención de datos y escritura. La parte de inicialización es básicamente donde se definen todas las variables, archivos que se van a abrir, y se inicializan algunas variables que sean necesarias. Se ha diseñado para que procesen un archivo de texto con un nombre indicado. En el caso de “ProcesaTCP.c”, procesará un archivo de texto con el nombre “TCP.txt”, mientras que

“ProcesaCTS.txt”, procesará un archivo de texto con el nombre “CTS.txt”. Antes de ejecutar ambos programas, hay que asegurarse previamente que ambos archivos de texto correspondientes a cada programa estén en el mismo directorio que el programa .c.

4.3.1. Lectura y extracción de datos

Los dos programas .C utilizan un sistema similar de funcionamiento. A través de un ciclo while, se leerá todo el archivo .txt obtenido de Wireshark. El ciclo terminará una vez se haya acabado el archivo. Con este ciclo while se irá leyendo línea a línea el archivo.

```
while(fscanf(myFile, "%[^\n]", input) != EOF)
```

Figura 4.16. Ciclo While para la lectura del archivo

A continuación se mostrará cómo se ha extraído los datos en cada programa. Todas las tramas de datos contenidas en los archivos de texto comparten una estructura, por lo tanto se puede analizar fácilmente, ya que de cada cierto número de líneas se puede extraer el dato que se desea. Se utilizarán condiciones if que funcionarán como marcadores para elegir la línea sobre la cual se van a extraer datos, haciendo que estos marcadores se vayan actualizando para conseguir todos los datos de todas las tramas del archivo de texto.

- ProcesaTCP.c

Este programa se encargará de procesar segmentos TCP. Todos los segmentos TCP guardados en el fichero de texto, vienen precedidas por un pequeño resumen del segmento en cuál aparecen datos interesantes como el tamaño del segmento.

Archivo	Edición	Formato	Ver	Ayuda
No.	Time	Source	Destination	Protocol Length Info
245132	38.495511	10.0.0.2	10.0.0.1	TCP 1574 53116 → 5201 [PSH, ACK] Seq=38 Ack=1 Win=64256 Len=1448

Figura 4.17. Resumen inicial de un segmento TCP

Conociendo el tamaño del segmento, se podrá extraer que segmentos son los que contienen los datos que son necesarios para el análisis. En este caso, habrá tres tamaños: 1579 Bytes, 1574 Bytes y 1567 Bytes. Con el comando sscanf se leerá la línea del archivo de texto y se extraerá tres datos: la longitud o tamaño del segmento, y un par de palabras para comprobar si se corresponde a una retransmisión TCP que se utilizarán más adelante. Los segmentos que interesan son los que tienen un tamaño de 1579 Bytes, ya que estos contienen la potencia de señal recibida del segmento y la velocidad de transmisión de la A-MPDU. También interesarán los segmentos de tamaño de 1574 Bytes ya que corresponden a segmentos que son transmitidos utilizando el estándar 802.11g. Estos segmentos de 1574 Bytes se originarán en escenarios con muy mala calidad en el enlace como puede ser el escenario de mayor distancia en el que transmisor y receptor están más alejados. Esta reducción de la tasa de transmisión la provoca el software del adaptador USB Wi-Fi, que para compensar la distancia mencionada, utiliza menores tasas de transmisión. El último tamaño de segmento es de 1567 Bytes y corresponde a los

segmentos contenidos en una A-MPDU. Estos segmentos son idénticos a los segmentos de 1579 Bytes, solamente que el segmento de 1579 Bytes contiene información de toda una ráfaga de segmentos contenidos en una A-MPDU. Por lo tanto, prácticamente se ignora este segmento, ya que todos los segmentos TCP de una misma ráfaga A-MPDU tienen la misma velocidad de transmisión, y con obtener el segmento de 1579 Bytes, ya se puede obtener toda la información necesaria de la ráfaga A-MPDU.

En la siguiente Figura 4.18, se podrá observar cómo lee la longitud del segmento y dependiendo de este, indicará dónde está, es decir, en qué línea del archivo de texto, el valor que indica la tasa de transmisión o en el caso de segmentos de 1567 Bytes, el identificador IP.

```
if (i==s){
    sscanf(input,"%*s %*s %*s %*s %*s %i %s %s",&longitud , TipoTCP1, TipoTCP2);

    if(longitud == 1579){
        ratecont = s+8;
    }else if(longitud == 1574){
        ratecont=s+6;
    } else{
        id = s + 20;
    }
}
```

Figura 4.18. Lectura del tamaño del segmento

En la Figura 4.19, se muestra como sscanf lee el dato correspondiente a la velocidad de transmisión, indicando además que dentro de tres líneas está el valor de la potencia de la señal recibida.

```
if (i == ratecont){
    Antena = ratecont +3;
    sscanf(input,"%*s %*s %f",&rate[j]);
}
```

Figura 4.19. Lectura de la velocidad de transmisión

En la siguiente Figura 4.20, se muestra como con sscanf se lee el dato correspondiente a la potencia de señal recibida, indicando además que dentro de 10 líneas está el valor del identificador IP.

```

if (i == Antena){
    sscanf(input,"%*s %*s %*s %i",&pot[j]);
    id= Antena + 10;
    j = j+1;
}

```

Figura 4.20. Lectura de la velocidad de transmisión

En la siguiente Figura, utilizando sscanf se lee el identificador de datagrama IP y además con un contador se registrará el número de segmentos TCP. La principal función de obtener el identificador IP o el número de segmentos es para solucionar problemas que surjan como una posible mala lectura. A continuación de esto último, hay una comparación tipo if, cuya función será determinar si el segmento leído es una retransmisión TCP (TCPRTX) o si es un segmento TCP que se ha transmitido desordenado (TCPOut). La razón de diferenciar este tipo de segmentos TCP del resto de segmentos TCP es debido a que Wireshark identifica que estos dos últimos tipos de segmento TCP, a pesar de tener un campo de datos, este está vacío. Al ser esta parte el final del ciclo while que se dedica a leer línea por línea, se deberá indicar en cuántas líneas comienza el siguiente segmento TCP. Al no tener campo de datos, en 11 líneas estará el siguiente segmento, mientras que el resto de los segmentos TCP tendrán que esperar la lectura de 103 líneas para pasar al siguiente segmento TCP.

```

if(i == id){
    sscanf(input,"%*s %*s %s",listaid);
    contrama= contrama +1;

    if(strcmp(TCPOut,TipoTCP2) == 0 || strcmp(TCPRTX,TipoTCP2) == 0){
        s=id+11;
    }else{
        s = id + 103;
    }
}
}

```

Figura 4.21. Lectura del identificador IP y comprobación si el segmento es una retransmisión TCP o un segmento TCP fuera de orden

- ProcesaCTS.c

Este programa se encargará de procesar tramas Clear-To-Send (CTS). El archivo de texto leído en este caso contendrá ramas de idéntica estructura, es decir, de un mismo tamaño, por lo que el análisis y extracción de datos será más sencilla en este caso. En este programa .c se extraerán dos tipos de tramas: el instante de tiempo en el que se capturó esa trama en concreto, y la cantidad de tiempo que se le otorga al transmisor para transmitir información. Cuanto más grande sea esta cantidad de tiempo, un mayor número de tramas podrá transmitir el transmisor/cliente.

Al igual que antes, antes de cada trama habrá un breve resumen sobre esta. En este caso de los datos que aparecerán, estará el instante de tiempo en el cual se ha capturado la trama. También he señalado como indica este resumen que la trama es tipo CTS.

CTS: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda
No.	Time	Source	Destination	Protocol Length Info
330	15.691536		Tp-LinkT_21:84:3a (64:70:02:21:84:3a) (RA)	802.11 52 Clear-to-send

Figura 4.22. Resumen inicial de una trama CTS

Para la lectura y extracción del instante del tiempo, se volverá a utilizar sscanf como en el programa .c, "ProcesaTCP.c". Se guardará el dato en un array y se indicará que 7 líneas después estará el tiempo otorgado al transmisor para poder enviar información.

```
if (i==s){
    sscanf(input,"%*s %f",&TiempoTrama[j]);
    Cts= s+7;
}
```

Figura 4.23. Lectura del instante de tiempo en que se captura la trama

El siguiente paso será volver a leer la línea indicada y esta vez se guardará en otro array el tiempo otorgado al transmisor indicado por la trama CTS. Una vez guardado, se indicará en cuántas líneas estará la siguiente trama CTS, que es 5 líneas.

```
if (i == Cts){
    sscanf(input,"%*s %*s %*s %*s %*s %*s %i",&CtsTiempo[j]);
    s= Cts + 5;
    j=j+1;
}
```

Figura 4.24. Lectura tiempo otorgado al transmisor

4.3.2. Escritura de resultados

En este apartado se comentará como se ha realizado la escritura de los datos deseados en un archivo de texto.

- ProcesaTCP.c

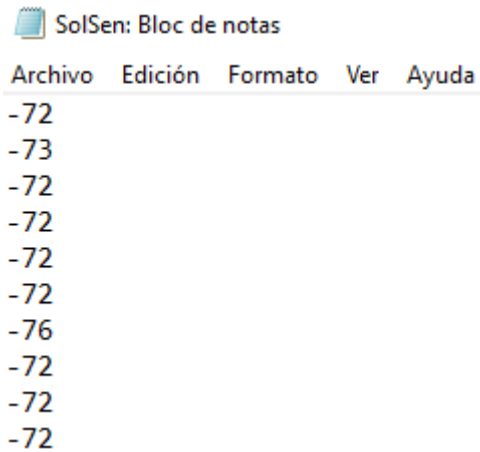
Primero se escribirá en el archivo de texto "SolSen.txt" todas las potencias de señal recibidas. Básicamente con un ciclo for se escribirá una lista en la que cada valor de la lista corresponderá con la potencia de señal recibida, en dBm, de cada ráfaga A-MPDU.

```

myFile2 = fopen("SolSen.txt", "w");
for(i=0;i<j;i++){
    fprintf(myFile2,"%i\n",pot[i]);
}
fclose(myFile2);

```

Figura 4.25. Escritura de los valores de potencia de señal recibida de cada ráfaga A-MPDU



```

SolSen: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
-72
-73
-72
-72
-72
-72
-72
-76
-72
-72
-72

```

Figura 4.26. Visualización del archivo "SolSen.txt"

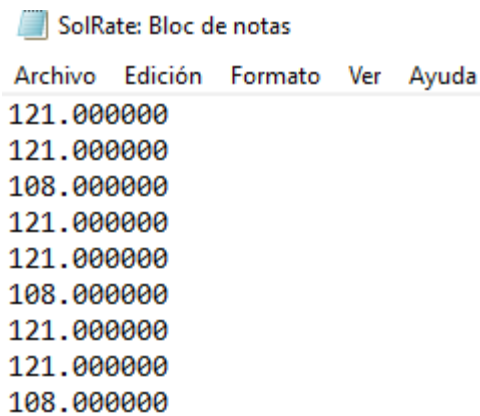
Se escribirá un segundo archivo de texto, "SolRate.txt", que contendrá las velocidades de transmisión, en Mbps, de cada ráfaga de segmentos A-MPDU. Al igual que antes, se escribirán los resultados en un archivo de texto utilizando un ciclo for.

```

myFile3 = fopen("SolRate.txt", "w");
for(i=0;i<j;i++){
    fprintf(myFile3,"%f\n",rate[i]);
}
fclose(myFile3);

```

Figura 4.27. Escritura de los valores de velocidad de transmisión de cada ráfaga A-MPDU



```

SolRate: Bloc de notas
Archivo Edición Formato Ver Ayuda
121.000000
121.000000
108.000000
121.000000
121.000000
108.000000
121.000000
121.000000
108.000000
108.000000

```

Figura 4.28. Visualización del archivo "SolRate.txt"

- ProcesaCTS.c

En este programa .c únicamente se escribirá un archivo, donde se escribirá el instante de tiempo de cada trama CTS y el tiempo otorgado al transmisor para enviar información indicado en cada trama CTS. Los datos de cada línea corresponderán a una trama CTS. Para escribir los datos se ha utilizado un ciclo for que vaya escribiendo en el archivo de texto "CTSprocesado.txt" tanto el instante de tiempo como el tiempo otorgado al transmisor a la vez de una misma trama CTS.

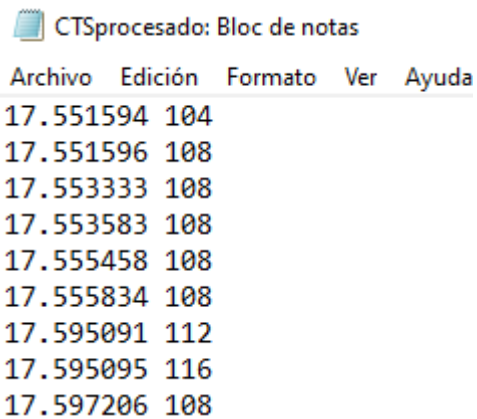
```

myFile2 = fopen("CTSprocesado.txt", "w");
for(i=0;i<j;i++){
    fprintf(myFile2,"%f %i\n",TiempoTrama[i], CtsTiempo[i]);
}
fclose(myFile2);

```

Figura 4.29. Escritura del instante de tiempo en segundos en que se recibió la trama CTS y la cantidad de tiempo que se le ha otorgado al transmisor para enviar información en microsegundos

El primer valor corresponde al instante de tiempo en que se ha recibido la trama en segundos, mientras que el segundo valor corresponde al tiempo otorgado al transmisor para enviar información en microsegundos. Se han escrito estos dos valores en un mismo archivo de texto ya que estos dos valores se van a utilizar en la misma gráfica más adelante en Matlab.



CTSPprocesado: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda
17.551594	104			
17.551596	108			
17.553333	108			
17.553583	108			
17.555458	108			
17.555834	108			
17.595091	112			
17.595095	116			
17.597206	108			

Figura 4.30. Visualización del archivo "CTSPprocesado.txt"

4.4. Matlab

Una vez extraídos los datos necesarios a través del programa .C y con los archivos de texto que los contienen, se procederá con Matlab a realizar una representación gráfica de los mismos. Para ello se utilizarán dos scripts: "dibuja.m" y "dibujaCTS". Para ejecutar el código Matlab simplemente se deberá llamar a la función Matlab con sus archivos de datos correspondientes como datos de entrada. En este caso sería: `dibuja("SolSen.txt", "SolRate.txt")` y `dibujaCTS("CTSPprocesado.txt")`. Ambos códigos Matlab estarán en un anexo.

- `dibuja.m`

En este primer script de Matlab, el primer paso será leer los datos. Primero se realizará la lectura del archivo "SolSen.txt". Con `formatSpec` se indicará que tipo de dato se está leyendo, en este caso un dato integer. Con `fscanf` se leerá el archivo y se guardará todos los valores de potencia recibida en un array "A". `LimSup` indica el número total de ráfagas A-MPDU.

```
fileID = fopen(x, 'r');
formatSpec = '%d';
A = fscanf(fileID, formatSpec);
LimSup=size(A);
```

Figura 4.31. Lectura datos del archivo SolSen.txt

A continuación, se representará estos datos en una gráfica. Donde el eje X indicará que ráfaga A-MPDU es y el eje Y indicará cuál es la potencia de señal recibida (RSSI) de esa A-MPDU en dBm.

```

figure(1)
plot(A)
xlim([0 LimSup(1)]);
ylabel('RSSI (dbm)')
xlabel('Número secuencia A-MPDU')
fclose(fileID);

```

Figura 4.32. Código para la creación de una gráfica que represente el RSSI

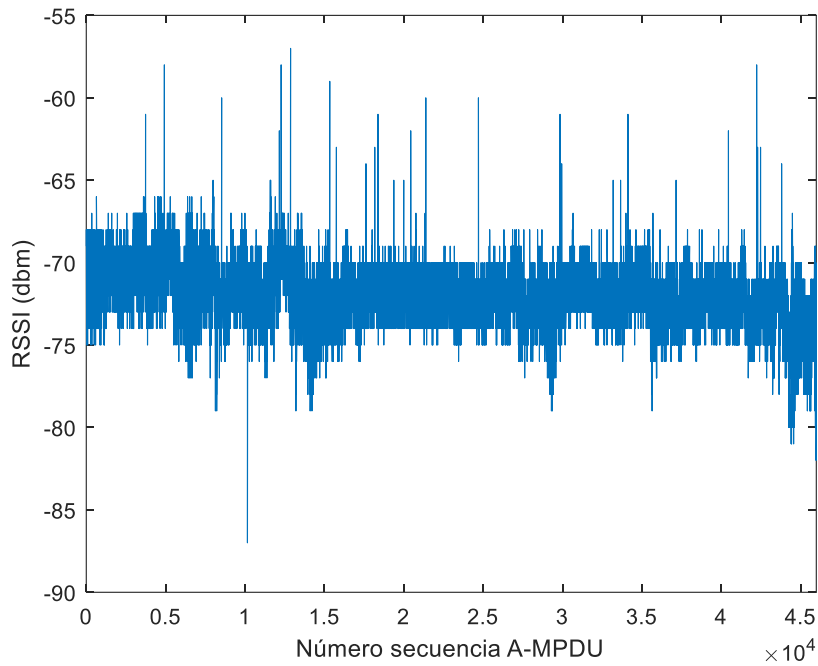


Figura 4.33. Representación gráfica del RSSI

El siguiente paso será la lectura del archivo de texto SolRate.txt, para guardar en un array las tasas de transmisión de cada ráfaga A-MPDU, siendo este valor de tipo float.

```

fileID2 = fopen(y, 'r');
formatSpec2 = '%f';
B = fscanf(fileID2, formatSpec2);
Tam=size(B);
Y=1:1:Tam;

```

Figura 4.34. Lectura datos del archivo SolRate.txt

A continuación, se representará la velocidad de transmisión de todas las ráfagas A-MPDU en una gráfica. Donde el eje X indicará que ráfaga A-MPDU es y el eje Y indicará cuál es la velocidad de transmisión de esa ráfaga A-MPDU en Mbps.

```

figure(2)
scatter(Y,B,'filled')
xlim([0 LimSup(1)])
ylabel('Velocidad transmisión (Mbps)')
xlabel('Número secuencia A-MPDU')
fclose(fileID2);

```

Figura 4.35. Código para la creación de una gráfica que represente las velocidades de transmisión de datos

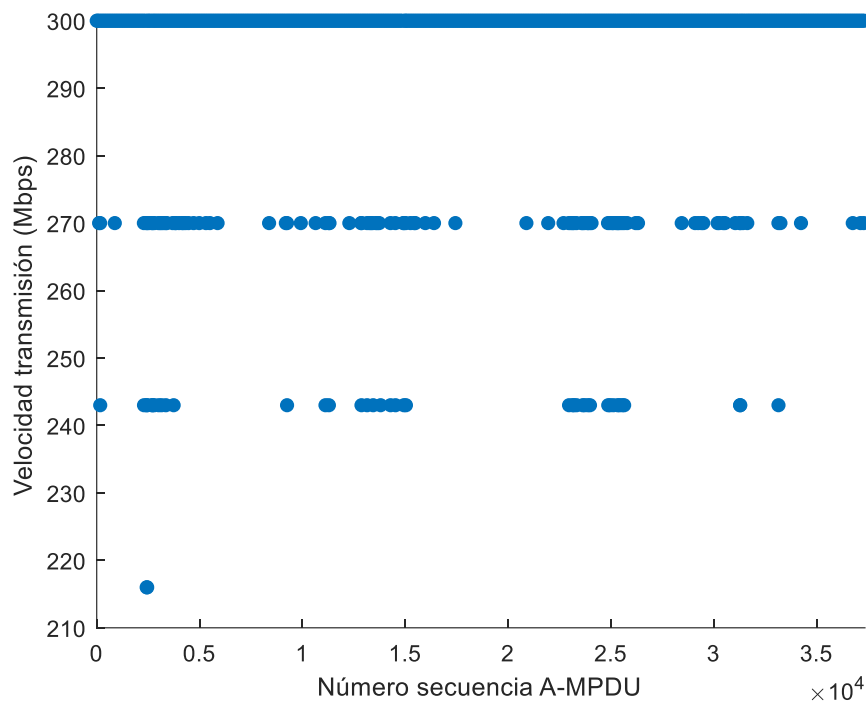


Figura 4.36. Representación gráfica de las velocidades de transmisión de datos

- dibujaCTS.m

En este código se realizará algo similar que la parte anterior, solo que ahora se leerán dos valores al mismo tiempo, el instante de tiempo en que se ha recibido la trama CTS y el tiempo que le ha sido al transmisor para poder enviar información. Estos datos se guardarán en una matriz de dos columnas.

```

fileID = fopen(x,'r');
formatSpec = '%f %d';
sizeA = [2 Inf];
A = fscanf(fileID,formatSpec,sizeA);
A=A';
Fin=size(A);

```

Figura 4.37. Lectura datos del archivo CTSprocesado.txt

Estos datos se representarán en una gráfica escalonada. Esta gráfica tendrá como eje X el momento del tiempo en que se ha capturado la trama CTS en segundos y en el eje Y estará el tiempo que le ha sido otorgado al transmitir para enviar datos en microsegundos.

```
figure(1)
LimInf=A(1,1)-2;
LimSup=A(Fin(1),1)+2;
stairs(A(:,1), A(:,2), 'LineWidth', 2)
ylabel('Duración CTS (microsegundos)')
xlabel('Tiempo (segundos)')
xlim([LimInf LimSup])
fclose(fileID);
```

Figura 4.38. Código para la creación de una gráfica que represente el instante de llegada de una trama CTS y cuánto tiempo le ha sido otorgado en esa trama CTS al transmisor para enviar información.

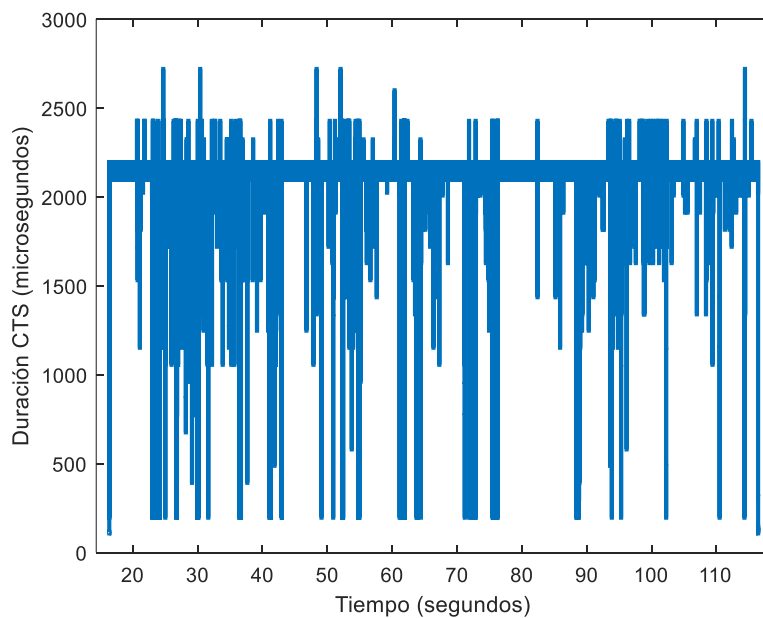


Figura 4.39. Representación gráfica del tiempo otorgado para enviar datos al transmisor

Capítulo 5. Análisis de los resultados

En este apartado se mostrarán todos los datos obtenidos de las medidas y se razonará el porqué de esos resultados, dependiendo del escenario, algoritmo de control de congestión y de si hay presencia de tráfico interferente dentro de la propia red Wi-Fi.

5.1. Beacon de la red Wi-Fi

En primer lugar, se mostrará una trama de “beacon” de la red Wi-Fi Ad-Hoc creada. Las tramas beacon de una red Wi-Fi contienen toda la información relacionada con la red inalámbrica y son difundidos periódicamente cada 100 ms (valor definido por el Beacon Interval, BI) para anunciar su presencia en un canal determinado. Al ser una red Ad-Hoc, los beacons los enviarán los propios dispositivos integrantes de la red Wi-Fi de forma distribuida.

Las tramas beacon vendrán identificadas en Wireshark como “Beacon Frame”. Se escogerá una trama beacon de la red Wi-Fi de trabajo que este caso se distinguirá por el nombre de la red Wi-Fi o SSID, en este caso, “mired40”.

1 0.000000	Tp-LinkT_21:6f:e2	Broadcast	802.11	171 Beacon frame, SN=171, FN=0, Flags=.....C, BI=100, SSID=mired40
2 0.001275	Tp-LinkT_21:84:3a	Broadcast	802.11	171 Beacon frame, SN=1892, FN=0, Flags=.....C, BI=100, SSID=mired40
3 0.102297	Tp-LinkT_21:6f:e2	Broadcast	802.11	171 Beacon frame, SN=172, FN=0, Flags=.....C, BI=100, SSID=mired40
4 0.103779	Tp-LinkT_21:84:3a	Broadcast	802.11	171 Beacon frame, SN=1893, FN=0, Flags=.....C, BI=100, SSID=mired40
5 0.204678	Tp-LinkT_21:6f:e2	Broadcast	802.11	171 Beacon frame, SN=173, FN=0, Flags=.....C, BI=100, SSID=mired40
6 0.206061	Tp-LinkT_21:84:3a	Broadcast	802.11	171 Beacon frame, SN=1894, FN=0, Flags=.....C, BI=100, SSID=mired40
7 0.307172	Tp-LinkT_21:6f:e2	Broadcast	802.11	171 Beacon frame, SN=174, FN=0, Flags=.....C, BI=100, SSID=mired40
8 0.308394	Tp-LinkT_21:84:3a	Broadcast	802.11	171 Beacon frame, SN=1895, FN=0, Flags=.....C, BI=100, SSID=mired40
9 0.409408	Tp-LinkT_21:6f:e2	Broadcast	802.11	171 Beacon frame, SN=175, FN=0, Flags=.....C, BI=100, SSID=mired40

Figura 5.1. Tramas Beacon de la red Wi-Fi Ad-Hoc creada en Wireshark

La trama beacon se transmitirá a la mínima velocidad posible (1 Mbps) utilizando el estándar 802.11b, ya que es la velocidad de transmisión menos restrictiva y por lo tanto un gran mayor número de redes Wi-Fi podrán percatarse de la existencia de la red Wi-Fi.

El campo que interesará de la trama Beacon es aquel donde se menciona las características de la red: IEEE 802.11 Wireless Management. En este campo se podrá encontrar características generales como velocidades de transmisión básicas soportadas (estándares 802.11b y 802.11g), tipo de seguridad que emplea, nombre de la red inalámbrica (SSID) y en qué canal se encuentra.

En la siguiente Figura, se puede observar el nombre de la red Wi-Fi o SSID, el tipo de red Wi-Fi, Ad-Hoc (IBSS), las tasas de transmisión que soporta, entre las que se incluyen las tasas de transmisión de los estándares 802.11b y 802.11g, ya que el estándar 802.11n tiene retrocompatibilidad con estándares antiguos, y en que canal se sitúa la red Wi-Fi, en este caso canal 3 (2422 MHz).

```

> Fixed parameters (12 bytes)
▼ Tagged parameters (93 bytes)
  > Tag: SSID parameter set: mired40
  > Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
  > Tag: DS Parameter set: Current Channel: 3
  > Tag: IBSS Parameter set: ATIM window 0x0
  > Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]

```

Figura 5.2. Información red Wi-Fi Ad-Hoc creada recogida en la trama de Beacon

Ya que se va a trabajar usando el estándar 802.11n, interesarán en especial dos campos, HT Capabilities (802.11n D1.10) y HT Information (802.11n D1.10), poniendo especial énfasis en el primero en que se muestra que características de 802.11n ofrece los adaptadores USB Wi-Fi. Cabe recordar, que a pesar de que se permita el uso de intervalo de guarda corto tanto en una red con canal de 20 MHz como en una red con canal de 40 MHz, solo se utilizará en canales de 40 MHz.

```

▼ HT Capabilities Info: 0x11ee
.... ..0 = HT LDPC coding capability: Transmitter does not support receiving LDPC coded packets
.... ..1 = HT Support channel width: Transmitter supports 20MHz and 40MHz operation
.... ..11.. = HT SM Power Save: SM Power Save disabled (0x3)
.... ..0.... = HT Green Field: Transmitter is not able to receive PPDU's with Green Field (GF) preamble
.... ..1.... = HT Short GI for 20MHz: Supported
.... ..1.... = HT Short GI for 40MHz: Supported
.... ..1.... = HT Tx STBC: Supported
.... ..01.... = HT Rx STBC: Rx support of one spatial stream (0x1)
.... ..0.... = HT Delayed Block ACK: Transmitter does not support HT-Delayed BlockAck
.... ..0.... = HT Max A-MSDU length: 3839 bytes
.... ..1.... = HT DSSS/CCK mode in 40MHz: Will/Can use DSSS/CCK in 40 MHz
.... ..0.... = HT PSMP Support: Won't/Can't support PSMP operation
.... ..0.... = HT Forty MHz Intolerant: Use of 40 MHz transmissions unrestricted/allowed
.... ..0.... = HT L-SIG TXOP Protection support: Not supported

```

Figura 5.3. Características opcionales de 802.11n de la red Wi-Fi creada recogida en la trama de Beacon

Además, se pueden extraer otros datos como por ejemplo el tamaño máximo de A-MPDU, que este caso es de 65535 Bytes. Como se observará más adelante, todas las ráfagas A-MPDU serán inferiores de tamaño a lo establecido en el beacon, por lo tanto se cumplirá. Otros datos que se pueden extraer es que la red Wi-Fi no permite ni el uso de Beamforming ni de selección de antenas. Por último, en el campo HT Information, se indicará si es un canal de 20 MHz o 40 MHz, y en el caso de que sea de 40 MHz, se indicará si el canal adicional secundario está en una frecuencia superior o inferior del canal principal.

```

▼ HT Information Subset (1 of 3): 0x05
.... ..01 = Secondary channel offset: Secondary channel is above the primary channel (0x1)
.... ..1.. = Supported channel width: Channel of any width supported
.... ..0... = Reduced Interframe Spacing (RIFS): Prohibited
0000 .... = Reserved: 0x0

```

Figura 5.4. Datos sobre el canal de una red Wi-Fi recogida en la trama de Beacon

Por último, en la pestaña “RX Supported Modulation and Coding Scheme Set” que se encuentra dentro de la pestaña “HT Capabilities” mencionada anteriormente, se puede encontrar el número de Spatial Streams que soporta los adaptadores USB Wi-Fi de la red Ad-Hoc creada, siendo en este caso, 2 spatial streams.

```

▼ Rx Supported Modulation and Coding Scheme Set: MCS Set
  > Rx Modulation and Coding Scheme (One bit per modulation): 2 spatial streams
    .... ..00 0000 0000 = Highest Supported Data Rate: 0x000
    .... ..1 = Tx Supported MCS Set: Defined
    .... ..0 = Tx and Rx MCS Set: Equal
    .... ..00.. = Maximum Number of Tx Spatial Streams Supported: 0x0, 2 spatial streams
    .... ..0 = Unequal Modulation: Not supported

```

Figura 5.5. N° de spatial streams de la red Wi-Fi Ad-Hoc de trabajo

5.2. Cálculo del throughput teórico

En este apartado se calculará el throughput teórico, es decir, la máxima tasa de transmisión de datos que podría alcanzarse en las medidas. Sólo se calculará el throughput teórico para el mejor caso, siendo el escenario de menor distancia donde mejores throughput se obtendrán. Este throughput teórico calculado se comparará con el throughput real obtenido. Se calculará el throughput tanto para un canal de 20 MHz como para uno de 40 MHz.

Para la transmisión de datos se sigue el esquema de la siguiente figura. El transmisor (source) deberá esperar un período de tiempo DIFS más un tiempo de Backoff para poder enviar un RTS (Request To Send). El receptor deberá esperar antes de enviarle el CTS (Clear To Send), un periodo de tiempo SIFS. Una vez al transmisor se le haya permitido transmitir al receptor, el transmisor esperará un periodo de tiempo SIFS antes de transmitir los datos utilizando una ráfaga A-MPDU. Por último, el receptor esperará un período de tiempo SIFS una vez el transmisor haya terminado de transmitir, y acto seguido el receptor confirmará con un Block Ack los datos recibidos en la A-MPDU.

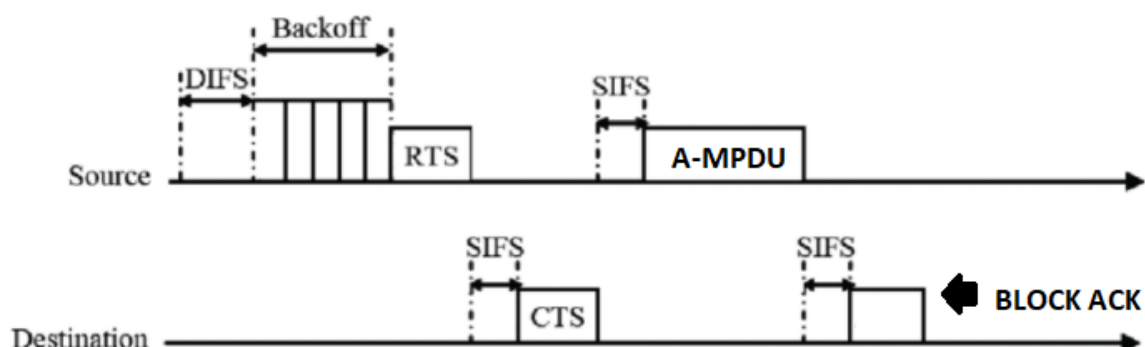


Figura 5.6. Esquema de acceso basado en intercambio RTS/CTS y Block ACK

Al estar utilizando TCP, también se deberá confirmar los segmentos TCP a nivel de capa TCP/IP ya que TCP es un protocolo orientado a la conexión. Por lo tanto, se repetirá el esquema anterior pero en dirección inversa. El receptor repetirá todo el proceso para poder enviar al transmisor una A-MPDU que en vez de datos contendrá confirmaciones ACK de la ráfaga A-MPDU de datos enviadas anteriormente.

Los datos para realizar los cálculos se extraerán de la siguiente Tabla 5.1. En esta tabla estarán los valores de tiempo mencionados anteriormente. Como se puede observar, en esta tabla no es solo que aparezcan varios estándares, sino que aparecen varias variaciones del estándar 802.11n sobre el cuál se va a trabajar. Estará marcado con un cuadro rojo la versión utilizada del estándar 802.11n, ya que se trabaja en la banda de frecuencias de 2.4 GHz y es compatible con estándares más antiguos. [9]

Tabla 5.1. Parámetros de la capa física y MAC de distintos estándares 802.11 [9]

Standard, rate	PLCP Preamble/Header	DIFS	SIFS	CW _{min}	CW _{max}	Slot Time
IEEE 802.11b, 1Mbps	192μs	50μs	10μs	31	1023	20μs
IEEE 802.11b, 2, 5.5, 11Mbps	96μs	50μs	10μs	31	1023	20μs
IEEE 802.11a	20μs	34μs	16μs	15	1023	9μs
IEEE 802.11g	20μs	28μs	10μs	15	1023	9μs
IEEE 802.11n HT Mixed, 5GHz	Non-HT preamble/header + 16μs + variable*	34μs	16μs	15	1023	9μs
IEEE 802.11n HT Greenfield, 5GHz	24μs + variable*	34μs	16μs	15	1023	9μs
IEEE 802.11n HT Mixed, 2.4GHz	Non-HT preamble/header + 16μs + variable*	28μs	10μs	15	1023	9μs
IEEE 802.11n HT Greenfield, 2.4GHz	24μs + variable*	28μs	10μs	15	1023	9μs
IEEE 802.11ac VHT	36μs + variable*	34μs	16μs	15	1023	9μs

Cada segmento de datos, en este caso cada A-MPDU, tiene un preámbulo. El tiempo que dura este preámbulo es un valor fijo. En este caso, será de 16 μs, más el preámbulo de un estándar de bajo rendimiento, siendo en este caso el preámbulo del estándar 802.11g, que dura 20 μs. Por último, quedará por definir una variable de tiempo que dependerá del número de spatial streams que utilice la red Wi-Fi. La ecuación para hallar ese tiempo variable es la siguiente:

$$T_{variable} = 4 * (N_{LTF} - 1) \quad [9]$$

N_{LTF} es el número de símbolos de largo entrenamiento y que depende directamente del número de spatial streams. En la red Wi-Fi de trabajo sólo habrá dos spatial streams. Para obtener ese valor se tendrá la siguiente tabla:

Tabla 5.2. Valor de N_{LTF} según el número de Spatial Streams [9]

Number of spatial streams, N_{SS}	IEEE 802.11n N_{LTF}
1	1
2	2
3	4
4	4

Al ser en este caso $N_{LTF} = 2$:

$$T_{variable} = 4 * (N_{LTF} - 1) = 4 * (2 - 1) = 4 \mu s$$

Por lo tanto el tiempo de preámbulo total será:

$$T_{preambulo} = T_{preambulo_802.11g} + 16 + T_{variable} = 20 + 16 + 4 = 40 \mu s$$

Las tramas CTS, RTS y Block Ack al utilizar el estándar 802.11g para transmitirse, tendrán un preámbulo de $20 \mu s$.

En esta versión del estándar 802.11n, el SIFS (Short Interframe Space) dura 10 microsegundos. El DIFS (DCF Interframe Space) ya se indica en la tabla 5.1 directamente que dura 28 microsegundos, pero también se podría calcular a mano, a partir del DIFS y del tiempo de slot (Slot Time), que aparece en la Tabla 5.1 y es de $9 \mu s$:

$$DIFS = SIFS + (2 * Slot Time) = 10 + (2 * 9) = 28 \mu s$$

Para calcular el tiempo de Backoff se deberá tener en cuenta que el Backoff es una variable aleatoria con distribución Uniforme con intervalo $[0, CW_{min} * Slot Time]$, donde el valor de CW_{min} se puede extraer de la Tabla 5.1 y es 15. El tiempo de Backoff será la media de esa variable aleatoria uniforme:

$$Media \text{ v. a. Uniforme}[x_1, x_2] = \frac{x_2 - x_1}{2}$$

$$Tiempo \text{ Backoff}[0, 15 * 9] = \frac{15 * 9 - 0}{2} = 67.5 \mu s$$

Por último, quedará por calcular el tiempo de duración de las distintas tramas o segmentos presentes en las medidas. En el caso de los segmentos TCP de datos tendrá un

tamaño de 1538 Bytes (Cabecera MAC 24 Bytes + QoS 2 Bytes + LLC/SNAP 8 Bytes + Trama IP 1500 Bytes + FCS 4 Bytes). Las tramas CTS tienen un tamaño de 20 Bytes, las tramas RTS tienen un tamaño de 14 Bytes, las tramas Block Ack tienen un tamaño de 32 Bytes y las tramas ACK de TCP tienen un tamaño de 90 Bytes.

El tiempo de estas tramas y segmentos se calcula de la siguiente manera, sin tener en cuenta los preámbulos:

$$T_{Trama}(\mu s) = \frac{Tamaño\ (Bytes) * 8}{Velocidad\ de\ transmisión}$$

Las velocidades de transmisión dependerán de estándar utilizado. Las tramas CTS, RTS y Block Ack al utilizar el estándar 802.11g se transmitirán a una velocidad de 24 Mbps, mientras que las A-MPDU se transmitirán a la máxima velocidad de transmisión posible dependiendo de la anchura del canal Wi-Fi. Para un canal de 20 MHz, la velocidad de transmisión será de 130 Mbps y para un canal de 40 MHz será de 300 Mbps.

Los tiempos resultantes, sin tener en cuenta los preámbulos, son los siguientes:

$$T_{RTS} = \frac{20 * 8\ bits}{24\ Mbps} = 6.6667\ \mu s$$

$$T_{CTS} = \frac{14 * 8\ bits}{24\ Mbps} = 4.6667\ \mu s$$

$$T_{Block\ Ack} = \frac{32 * 8\ bits}{24\ Mbps} = 10.6667\ \mu s$$

Para un canal de 20 MHz:

$$T_{Trama\ Datos} = \frac{1538 * 8\ bits}{130\ Mbps} = 94.6462\ \mu s$$

$$T_{Trama\ ACK} = \frac{90 * 8\ bits}{130\ Mbps} = 5.5385\ \mu s$$

Para un canal de 40 MHz:

$$T_{Trama\ Datos} = \frac{1538 * 8\ bits}{300\ Mbps} = 41.0133\ \mu s$$

$$T_{Trama\ ACK} = \frac{90 * 8\ bits}{300\ Mbps} = 2.4\ \mu s$$

Debido a que la duración de tiempo de símbolo es un valor fijo, se deberán ajustar estos valores de tiempos para que sean múltiplo de este tiempo de símbolo. El tiempo de símbolo en el canal de 20 MHz es 4 μs porque tiene el intervalo de guarda largo activado, por lo que las subportadoras OFDM estarán separadas 800 ns. En el caso del canal de 40 MHz, como se utiliza el intervalo de guarda corto, las subportadoras OFDM estarán separadas 400 ns y el tiempo de símbolo será de 3.6 μs . Todas las tramas de control RTS, CTS y Block Ack al estar utilizando el estándar 802.11g para transmitirlos, utilizarán intervalo de guarda largo y por lo tanto en esos casos el tiempo de símbolo es 4 μs . [9]

Comenzando con las tramas de control, si se redondea el valor obtenido previamente a un múltiplo de 4 (tiempo de símbolo) y se le suma el preámbulo que al ser estándar 802.11g es de 20 μs , se obtienen los siguientes resultados:

$$T_{RTS_Final} = Preambulo + Redondeo a múltiplo de 4 (T_{RTS}) = 20 + 8 = 28 \mu s$$

$$T_{CTS_Final} = 20 + 8 = 28 \mu s$$

$$T_{BlockAck_Final} = 20 + 12 = 32 \mu s$$

El siguiente paso será hallar el tiempo real de cada trama (MPDU) que esté contenida dentro de cada A-MPDU. Cabe recordar, que este caso no se incluirá el preámbulo, debido a que solo lleva preámbulo la A-MPDU al completo, no cada trama contenida en ella. Los cálculos resultantes después del redondeo a un valor múltiplo del tiempo de símbolo son los siguientes:

Tabla 5.3. Tiempos reales de las tramas MPDU contenidas dentro de una A-MPDU

	Canal de 20 MHz	Canal de 40 MHz
Tiempo de símbolo (μs)	4	3.6
Tiempo Trama MPDU Datos (μs)	96	43.2
Tiempo Trama MPDU Ack (μs)	8	3.6

A continuación, se calcularía el tiempo total de ráfaga A-MPDU. Lo lógico sería pensar que simplemente sería coger el tiempo obtenido en la tabla 5.3 y multiplicarlo por el número de tramas MPDU y añadirle el preámbulo de 40 μs del estándar 802.11n obtenido previamente. Pero hay que tener en cuenta una cosa que es que entre cada trama MPDU o trama agregada dentro de una ráfaga A-MPDU, hay una separación de 4 Bytes para delimitar las diferentes tramas MPDU. Al ser un tamaño tan pequeño, y al ocupar un período de símbolo, siempre el envío de estos 4 Bytes durará lo mismo que el tiempo de símbolo, que es el menor tiempo posible.

El número de tramas agregadas en una ráfaga A-MPDU puede variar, siendo el valor máximo posible aquel que se vio en el Beacon Frame del apartado 5.1. Aquí el tamaño de la A-MPDU vendrá delimitado por el tiempo que se otorgue para transmitir, por lo que nunca se alcanzará ese límite del tamaño. En este caso, en el escenario de menor distancia, la gran mayoría de las ráfagas A-MPDU tendrá un tamaño de 22 tramas agregadas en una A-MPDU.

Los tiempos de A-MPDU de datos o Ack calculados son los siguientes:

Para un canal de 20 MHz:

$$T_{A-MPDU \text{ Datos}} = \text{Preambulo} + N^{\circ}MPDUs * T_{MPDU \text{ Datos}} + (N^{\circ}MPDUs - 1) * T_{Símbolo}$$

$$= 40 + 22 * 96 + 21 * 4 = 2236 \mu s$$

$$T_{A-MPDU \text{ ACK}} = 40 + 22 * 8 + 21 * 4 = 300 \mu s$$

Para un canal de 40 MHz:

$$T_{A-MPDU \text{ Datos}} = 40 + 22 * 43.2 + 21 * 3.6 = 1066 \mu s$$

$$T_{A-MPDU \text{ ACK}} = 40 + 22 * 3.6 + 21 * 3.6 = 194,8 \mu s$$

Por último, quedará por calcular la cantidad de datos que se envían por ráfaga A-MPDU. Tal y como se indicó a la hora de realizar las medidas con iperf3, se especificó que cada trama MPDU debería tener un campo de datos TCP de 1448 Bytes, por lo tanto, la cantidad de bits que se envían en una ráfaga A-MPDU de 22 MPDUs es la siguiente:

$$L_{A-MPDU} = L_{MPDU} * 8 * N^{\circ}MPDUs = 1448 * 8 * 22 = 254848 \text{ bits}$$

A continuación, se mostrará una tabla con todos los resultados de los cálculos obtenidos. Con estos valores se calculará el valor del throughput teórico para un canal de 20 MHz y para un canal de 40 MHz.

Tabla 5.4. Valores de los parámetros de capa física y MAC

DIFS	28 μs
Tiempo Backoff	67,5 μs
SIFS	10 μs
$T_{RTS \text{ Final}}$	28 μs
$T_{CTS \text{ Final}}$	28 μs
$T_{BlockAck \text{ Final}}$	32 μs
L_{A-MPDU}	254848 bits
$T_{A-MPDU \text{ Datos}}$ (Canal de 20 MHz)	2236 μs
$T_{A-MPDU \text{ ACK}}$ (Canal de 20 MHz)	300 μs
$T_{A-MPDU \text{ Datos}}$ (Canal de 40 MHz)	1066 μs
$T_{A-MPDU \text{ ACK}}$ (Canal de 40 MHz)	194,8 μs

Siguiendo el esquema de la Figura 5.6, se calculará el throughput teórico. Se utilizarán las siguientes fórmulas. La primera expresión permite obtener el tiempo total que tarda una

ráfaga A-MPDU en transmitirse y ser reconocida (ACK) por el receptor. La segunda fórmula es el cálculo del throughput teórico:

$$T_{TX} = 2 * DIFS + 2 * Tiempo Backoff + 2 * T_{RTS_{Final}} + 2 * T_{CTS_{Final}} + 2 * T_{BlockAck_{Final}} + 6 * SIFS + T_{A-MPDU\ Datos} + T_{A-MPDU\ ACK}$$

$$Thput_{Teorico} = \frac{L_{A-MPDU}}{T_{TX}}$$

Para un canal de 20 MHz:

$$T_{TX} = 2 * 28 + 2 * 67.5 + 2 * 28 + 2 * 28 + 2 * 32 + 6 * 10 + 2236 + 300 = 2963 \mu s$$

$$Thput_{Teorico} = \frac{254848 \text{ bits}}{2963 \mu s} = 86.0101 \text{ Mbps}$$

Para un canal de 40 MHz:

$$T_{TX} = 2 * 28 + 2 * 67.5 + 2 * 28 + 2 * 28 + 2 * 32 + 6 * 10 + 1066 + 194.8 = 1687.8 \mu s$$

$$Thput_{Teorico} = \frac{254848 \text{ bits}}{1687.8 \mu s} = 150.9942 \text{ Mbps}$$

A continuación, se mostrará en una tabla, la comparación entre en throughput teórico calculado y el throughput real medido en un entorno real.

Tabla 5.5. Throughput teórico vs. Throughput real

	Throughput Teórico (Mbps)	Throughput Real (Mbps)
Canal de 20 MHz	86.0101	82.8
Canal de 40 MHz	150.9942	131

Las medidas de throughput obtenidas de un entorno real representan las mejores medidas obtenidas, perteneciendo ambas medidas al escenario de menor distancia como ya se mencionó antes. Como es de esperar, el throughput teórico es superior al throughput real. Esto es debido principalmente a dos razones: la primera es que aunque la mayoría de las ráfagas A-MPDU tienen 22 tramas agregadas, no es raro ver alguna A-MPDU con un menor número. Hay veces que incluso, cuando se comienza la medida, que la primera ráfaga A-MPDU comienza con 8 tramas agregadas. Otro problema viene en las confirmaciones ACK de TCP. Aquí en los cálculos se ha asumido que si la ráfaga A-MPDU de datos es de 22 tramas agregadas, la ráfaga A-MPDU de tramas ACK debería ser del mismo tamaño, pero comprobando las medidas en Wireshark, no es así, sino que suelen ser ráfagas de 10 o más tramas agregadas en una A-MPDU. Por último, no se han tenido en cuenta las retransmisiones de paquetes. A pesar de que al ser las medidas pertenecientes al escenario de menor distancia, se producen retransmisiones de tramas a nivel de capa MAC, lo cual reduce la tasa de transmisión efectiva.

Como se ha podido observar, cuanto mayor sea el número de tramas agregadas dentro de una A-MPDU, mayor será la tasa de transmisión efectiva o throughput. Una de las ventajas del uso de A-MPDU, es que permite enviar una gran cantidad de tramas que compartirán cabecera física, transmitiendo un único preámbulo antes de cada A-MPDU, y permitiendo el reconocimiento en bloque de todas esas tramas agregadas dentro de una A-MPDU utilizando el Block Ack.

A continuación, se mostrarán dos gráficas que mostrarán la evolución del throughput teórico según el número de tramas agregadas en una A-MPDU. El mejor valor de throughput teórico que aparece en las gráficas corresponderá al calculado en este documento, que es para una A-MPDU con 22 tramas agregadas. Por último, se indicará con una línea roja el throughput real obtenido en las medidas. Todos los puntos que estén por encima de esa línea roja indicarían que tendrían un mejor rendimiento que en un entorno real. Aquel punto de la gráfica que esté más cercano a la línea representaría el valor de throughput teórico más similar al throughput real de las medidas.

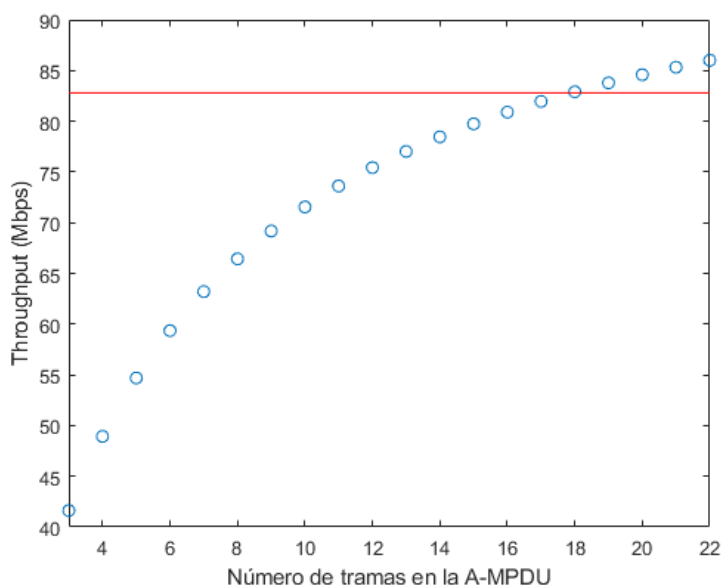


Figura 5.7. Evolución throughput teórico para un canal de 20 MHz

En la Figura 5.7, se puede observar cómo a partir de 18 segmentos agregados en una A-MPDU, el throughput teórico en un canal de 20 MHz ya tiene un rendimiento mejor que el throughput medido en un entorno real. También se puede obtener el dato, de que el throughput real obtenido de la medida es muy similar al throughput teórico de una A-MPDU con 18 segmentos agregados.

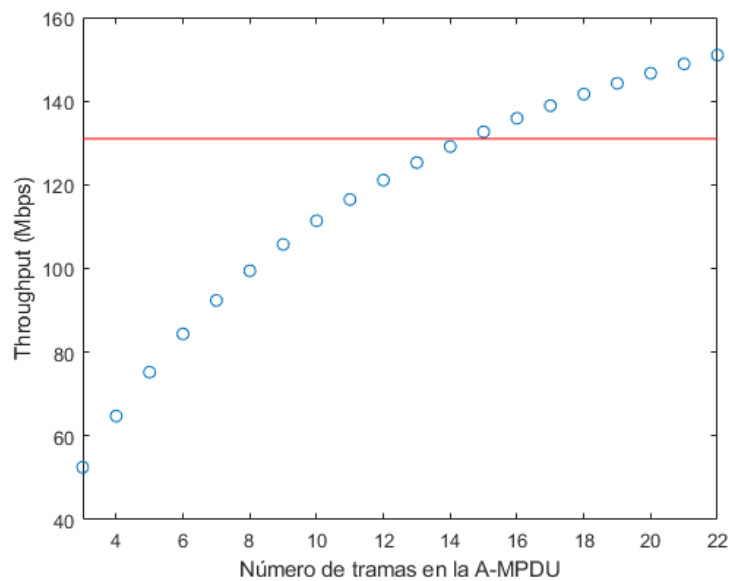


Figura 5.8. Evolución throughput teórico para un canal de 40 MHz

En la Figura 5.8, se puede observar cómo en un canal de 40 MHz a partir de 15 segmentos agregados en una A-MPDU, el throughput teórico ya tiene un rendimiento mejor que el throughput medido en un entorno real. El throughput real obtenido de la medida es muy similar al throughput teórico de una A-MPDU con 15 tramas agregadas.

Si se hiciese una comparativa frente al canal de 20 MHz, el canal de 20 MHz es mucho más eficiente, ya que su valor de throughput real es mucho más cercano al valor de throughput teórico, solamente 5 Mbps de diferencia, mientras que en el canal de 40 MHz hay 20 Mbps de diferencia, para 22 tramas agregadas en una A-MPDU. También está el dato que el canal de 40 MHz a pesar de trabajar a la máxima velocidad de transmisión, 300 Mbps, el throughput real de la medida indica como si trabajase con una A-MPDU que solo contiene 15 tramas agregadas, lo cual es 3 menos tramas agregadas que en un canal de 20 MHz. La causa de pérdida de efectividad de un canal de 40 MHz es debido a que a pesar del aumento de las tasas de transmisión, de 130 Mbps a 300 Mbps, el número de retransmisiones aumenta cuanto mayor sea la tasa de retransmisión. Además de que al trabajar a altas tasas de retransmisión, se es más susceptible a cambios de la potencia y a qué el mecanismo de adaptación de la tasa binaria de los adaptadores USB Wi-Fi lo detecte y reduzca ligeramente la tasa de transmisión. [10][11] A pesar de todo esto, no hay que olvidar que al usar un canal de 40 MHz, se ha pasado de un throughput real de 82.8 Mbps (Canal de 20 MHz) a un throughput de 131 Mbps (Canal de 40 MHz), una mejora de 48.2 Mbps. Por lo tanto, a pesar de ser el canal de 40 MHz menos eficiente que un canal de 20 MHz, sigue siendo una grandísima mejora en términos de throughput respecto al canal de 20 MHz.

5.3. Análisis de los resultados de las medidas

En este apartado se mostrarán los resultados obtenidos de las medidas iperf3 de 100 segundos cada una. La mayoría de las gráficas se han obtenido procesando las capturas de datos

de Wireshark y utilizando los programas siguiendo el procedimiento del apartado 4. Como ya se mencionó, habrá tres escenarios: de menor distancia, distancia media y de mayor distancia.

5.3.1. Escenario de menor distancia

En este escenario tanto transmisor y receptor estarán próximos a una mínima distancia. Para ver más información sobre el escenario de menor distancia consultar el apartado 3.2.1. Las medidas se realizarán en dos canales de distinta anchura: 20 MHz y 40 MHz.

- Canal de 20 MHz

Las medidas realizadas se dividirán en dos tipos: sin tráfico interferente o con tráfico interferente. Con tráfico interferente se hace referencia a que se crearán flujos de datos dentro de la misma celda de la red Ad-Hoc en la cual se van a realizar las medidas. Para más información sobre el esquema y disposición de los ordenadores que generan tráfico interferente consultar el apartado 3.2.

- Sin Tráfico Interferente

En este caso, en el escenario de menor distancia, no habrá ningún tipo de tráfico interferente dentro de la propia red Ad-Hoc. Dependiendo de si se ha utilizado el algoritmo de control de congestión New Reno o Cubic, los resultados de las medidas difieren.

- ❖ Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia sin tráfico interferente empleando el algoritmo de control de congestión Cubic para un canal de 20 MHz. Primero se mostrará en una tabla el throughput total de la medida y el número de retransmisiones que se han producido y de qué tipo son. Cabe recordar que las “fast retransmission” son retransmisiones TCP producidas por un triple ACK, en las cuales la ventana de congestión no se reduce drásticamente, mientras que las retransmisiones TCP por time-out sí que producen una reducción drástica del rendimiento. También en la tabla se indicará que cantidad de datos se ha enviado en esa medida de 100 segundos.

Tabla 5.6. Resultados medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	82.4
Volumen datos transmitidos (MBytes)	982
Retransmisiones	0

Al ser el escenario de menor distancia, las medidas realizadas son muy buenas y no hay retransmisiones TCP. A continuación, se mostrarán los resultados obtenidos gráficamente de la medida. La primera gráfica representará la evolución de la tasa de transmisión efectiva, throughput, en Mbps. Este valor se obtiene directamente de los resultados de la medida otorgados por la herramienta de medida Iperf3, que indica cada segundo de la medida el throughput medio.

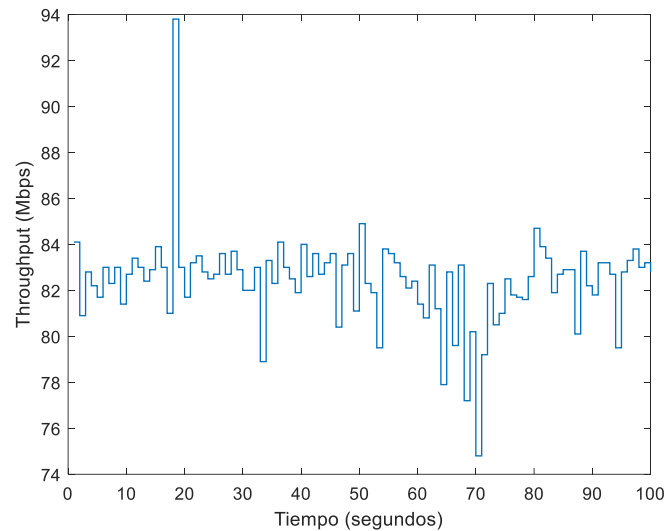


Figura 5.9. Evolución del throughput durante la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

La siguiente gráfica muestra la evolución de la ventana de congestión a lo largo del tiempo. Este valor también se obtiene de los resultados de la herramienta de medida iperf3.

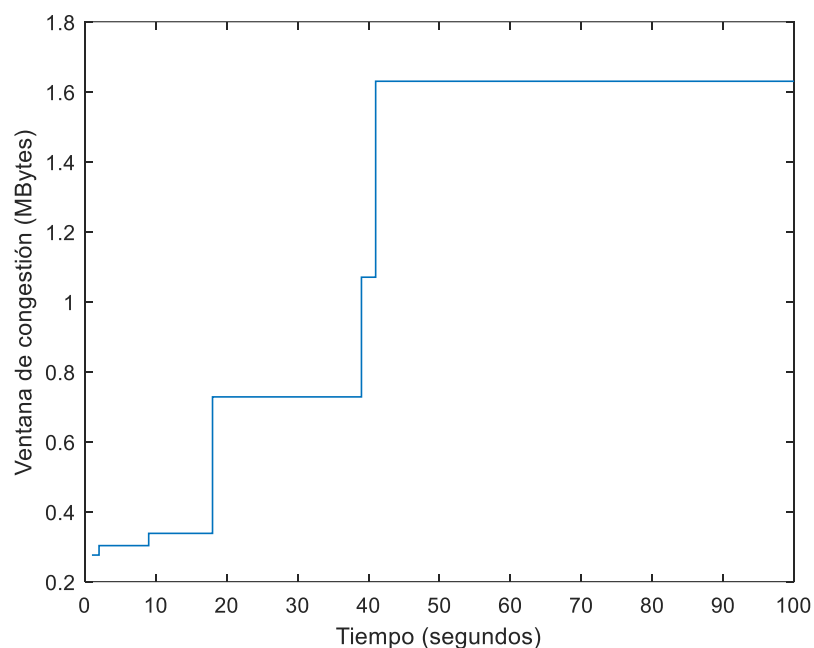


Figura 5.10. Evolución de la ventana de congestión durante la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

La siguiente gráfica muestra la potencia de recepción de señal en el receptor.

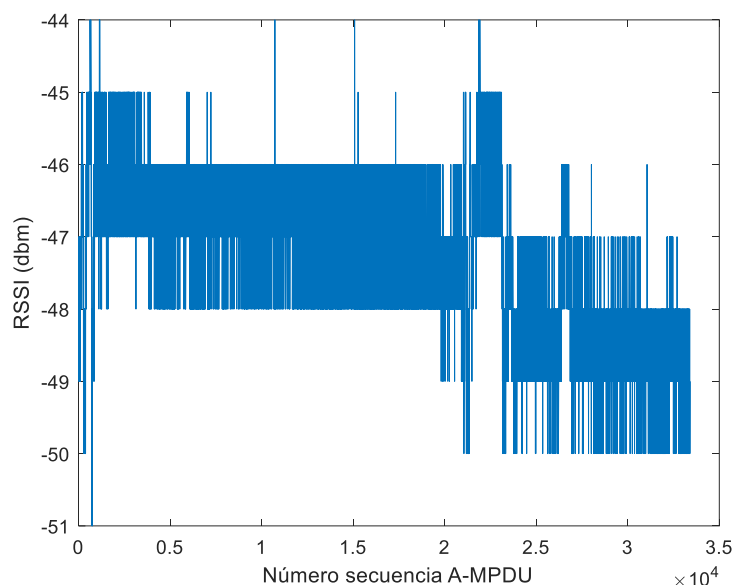


Figura 5.11. Potencia de señal recibida en el receptor en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

A continuación, se mostrará una gráfica que indicará el valor de la velocidad de transmisión de las ráfagas A-MPDU.

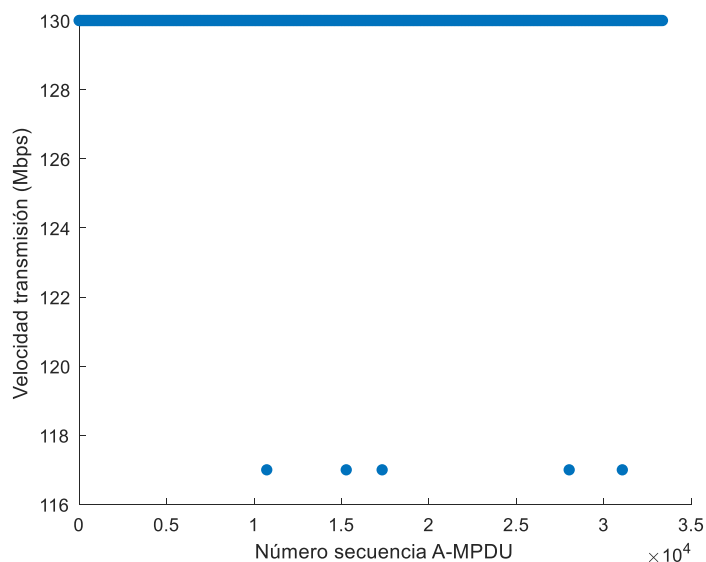


Figura 5.12. Velocidad de transmisión en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

La última gráfica que se enseñará indicará los tiempos que son otorgados al transmisor para poder enviar información. Cuanto mayor sea este tiempo, mayor será el número de tramas

agregadas que contenga la A-MPDU, por lo tanto, más información enviara. Como se puede observar, la gráfica empieza en el eje X un poco después del segundo 0, esto es debido a que está indicando el tiempo que Wireshark lleva capturando, ya que la medida no se realiza al instante, sobre todo en el escenario de mayor distancia, con un retardo de hasta 60 segundos.

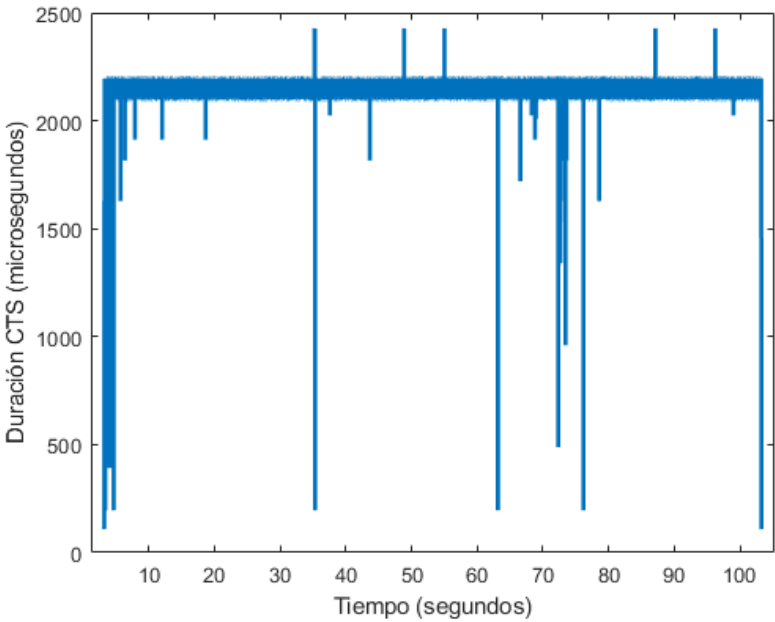


Figura 5.13. Tiempo disponible para transmitir del transmisor en la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

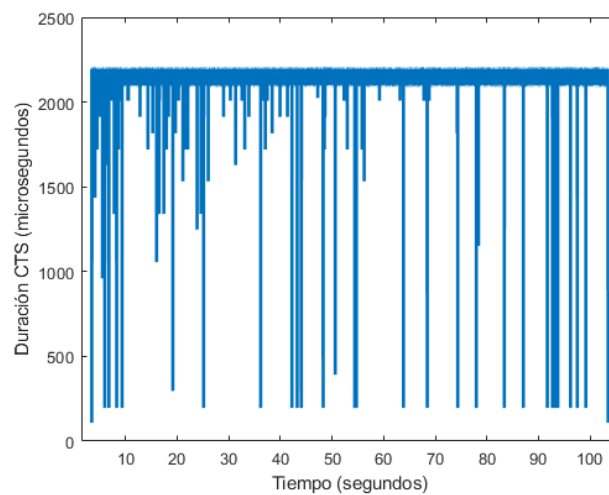
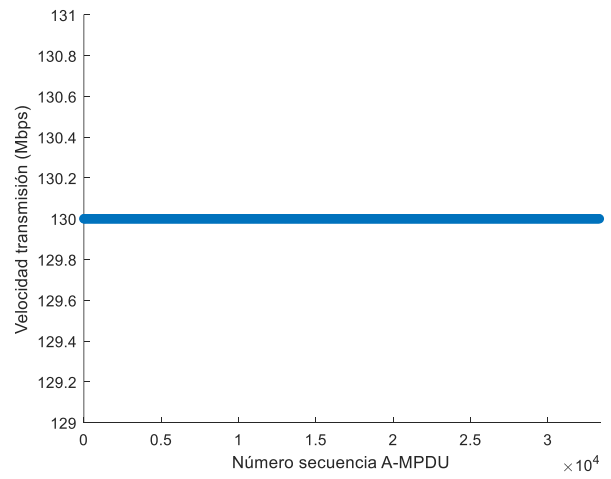
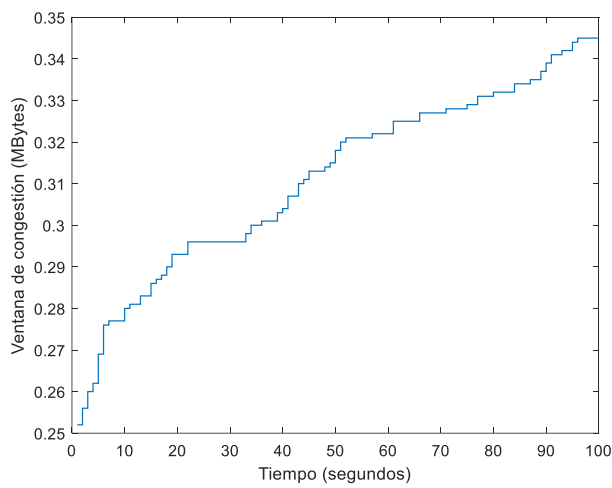
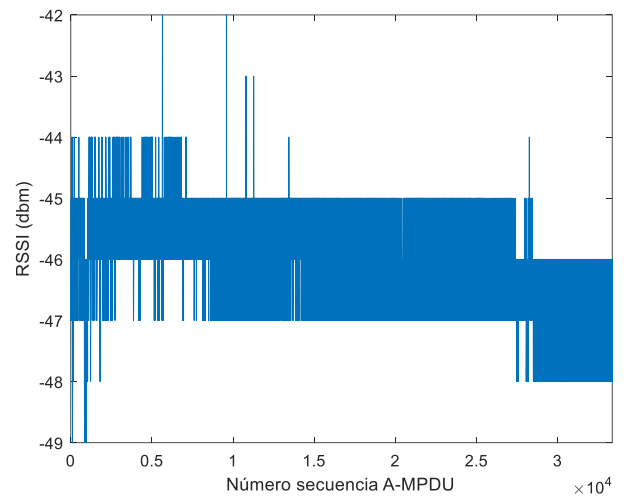
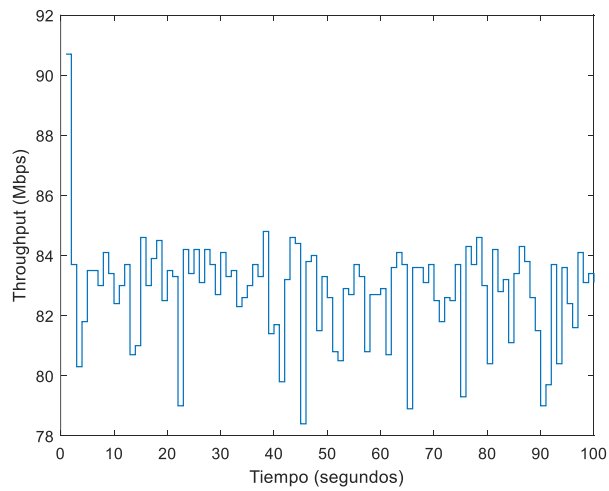
Como se puede observar, los resultados de esta medida son muy buenos, ya que como indica la Figura 5.12, se ha estado transmitiendo a máxima velocidad de transmisión prácticamente todo el tiempo y no ha habido ningún tipo de retransmisión.

- Canal de 20 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia sin tráfico interferente empleando el algoritmo de control de congestión New Reno para un canal de 20 MHz.

Tabla 5.7. Resultados medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	82.8
Volumen datos transmitidos (MBytes)	987
Retransmisiones	0



Figuras 5.14. Resultados de la medida del escenario de menor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno

Esta medida al igual que la anterior son prácticamente similares. El utilizar como algoritmo de control de congestión New Reno simplemente ha modificado la evolución de la ventana de congestión, teniendo esta un crecimiento más gradual.

- Canal de 20 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia con tráfico interferente empleando el algoritmo de control de congestión Cubic para un canal de 20 MHz.

Tabla 5.8. Resultados medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	29.6
Volumen datos transmitidos (MBytes)	353
Fast Retransmisión (Triple ACK)	1
Retransmisión por Time-Out	13

La siguiente gráfica de la Figura 5.15, representa el número de retransmisiones que se ha producido en cada instante de tiempo durante la medida. La razón por la que aparece esta gráfica por primera vez en este subapartado es debido que en las anteriores medidas no había retransmisiones. Si se suman todas las retransmisiones que aparecen en la gráfica, el resultado debería ser el marcado en la tabla anterior, en este caso, 14 retransmisiones.

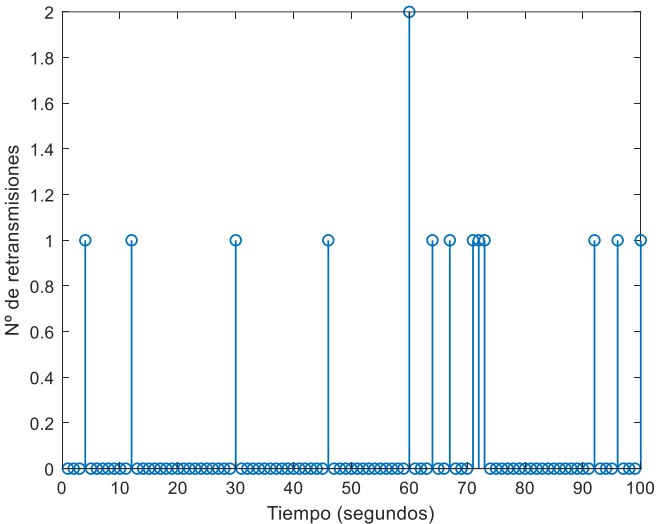
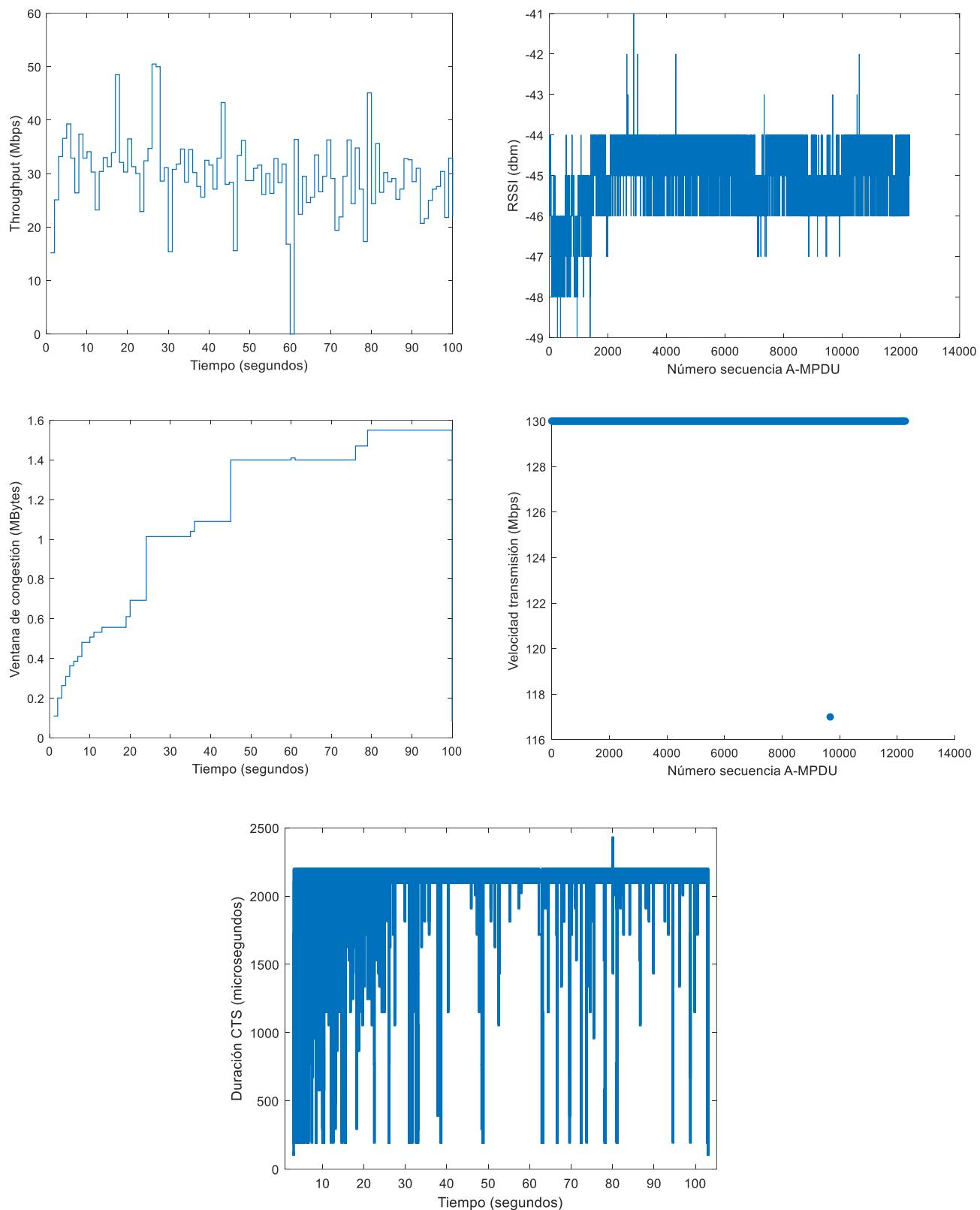


Figura 5.15. Número de retransmisiones en cada instante de tiempo en la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic



Figuras 5.16. Resultados de la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando Cubic

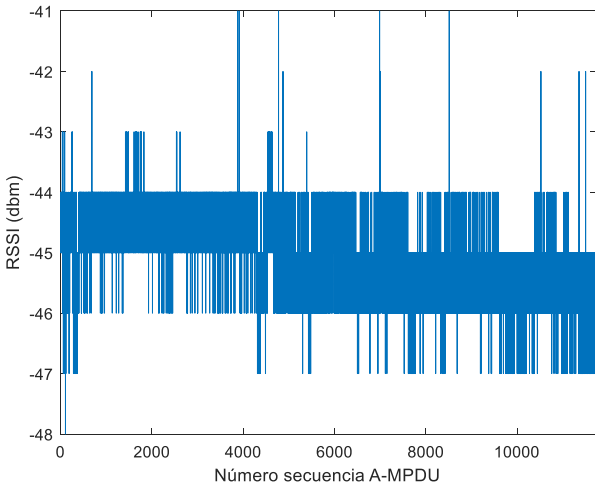
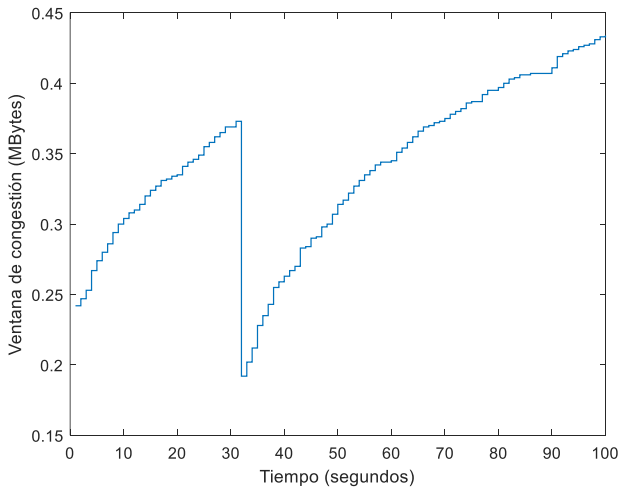
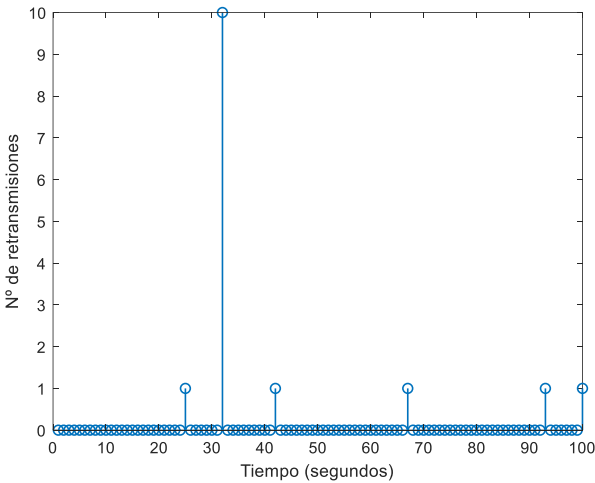
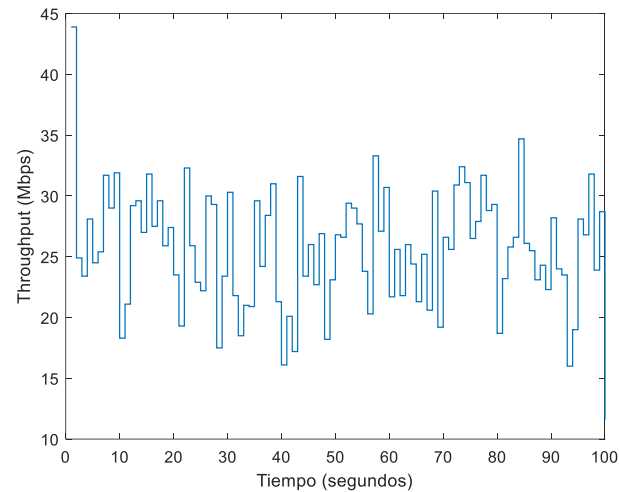
Si se compara esta medida con su contrapartida, que es la medida sin tráfico interferente utilizando Cubic, se puede observar como a pesar de que las velocidades de transmisión son prácticamente las mismas, el throughput se ha reducido drásticamente a 30 Mbps de media. Esto es debido a la aparición de retransmisiones TCP y a que está menos tiempo transmitiendo debido a que debe turnarse con los otros dispositivos de la red Wi-Fi para poder transmitir.

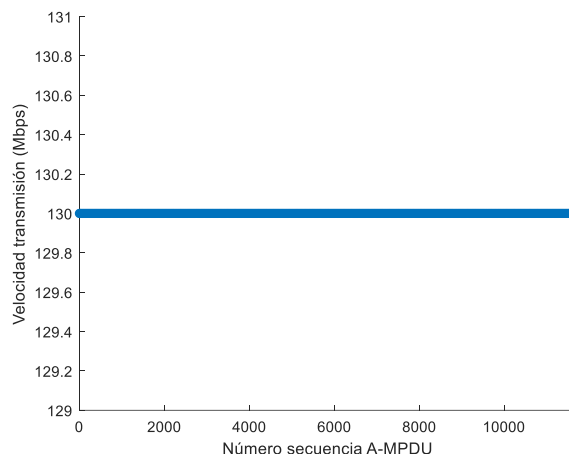
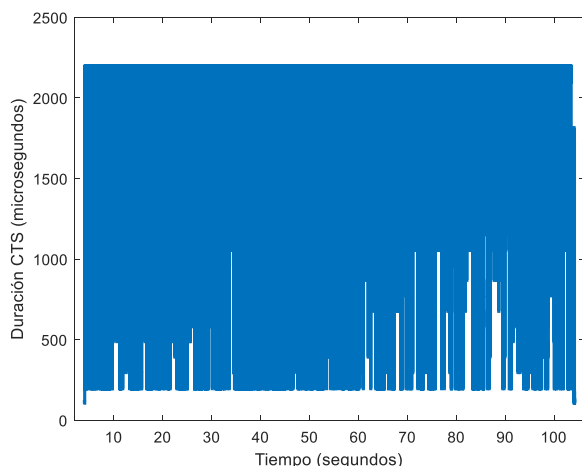
- Canal de 20 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia con tráfico interferente empleando el algoritmo de control de congestión New Reno para un canal de 20 MHz.

Tabla 5.9. Resultados medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	25.6
Volumen datos transmitidos (MBytes)	305
Fast Retransmisión (Triple ACK)	0
Retransmisión por Time-Out	15





Figuras 5.17. Resultados de la medida del escenario de menor distancia canal de 20 MHz con tráfico interferente utilizando New Reno

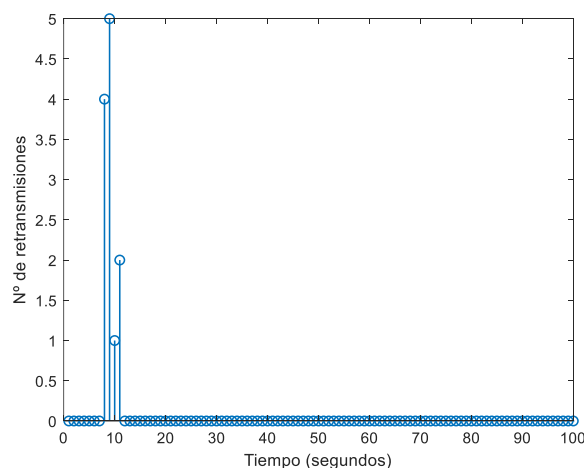
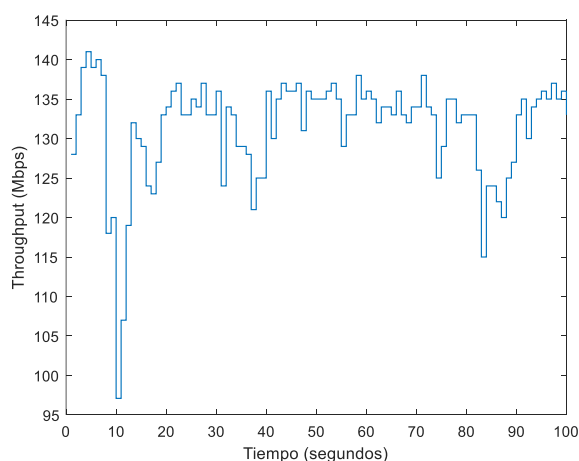
En este caso, al igual que en Cubic se produce un descenso del throughput por razones ya explicadas. La principal diferencia otra vez será la evolución de la ventana congestión. El que la principal diferencia entre las medidas de Cubic y New Reno sea la ventana de congestión será una constante en la mayoría de las medidas.

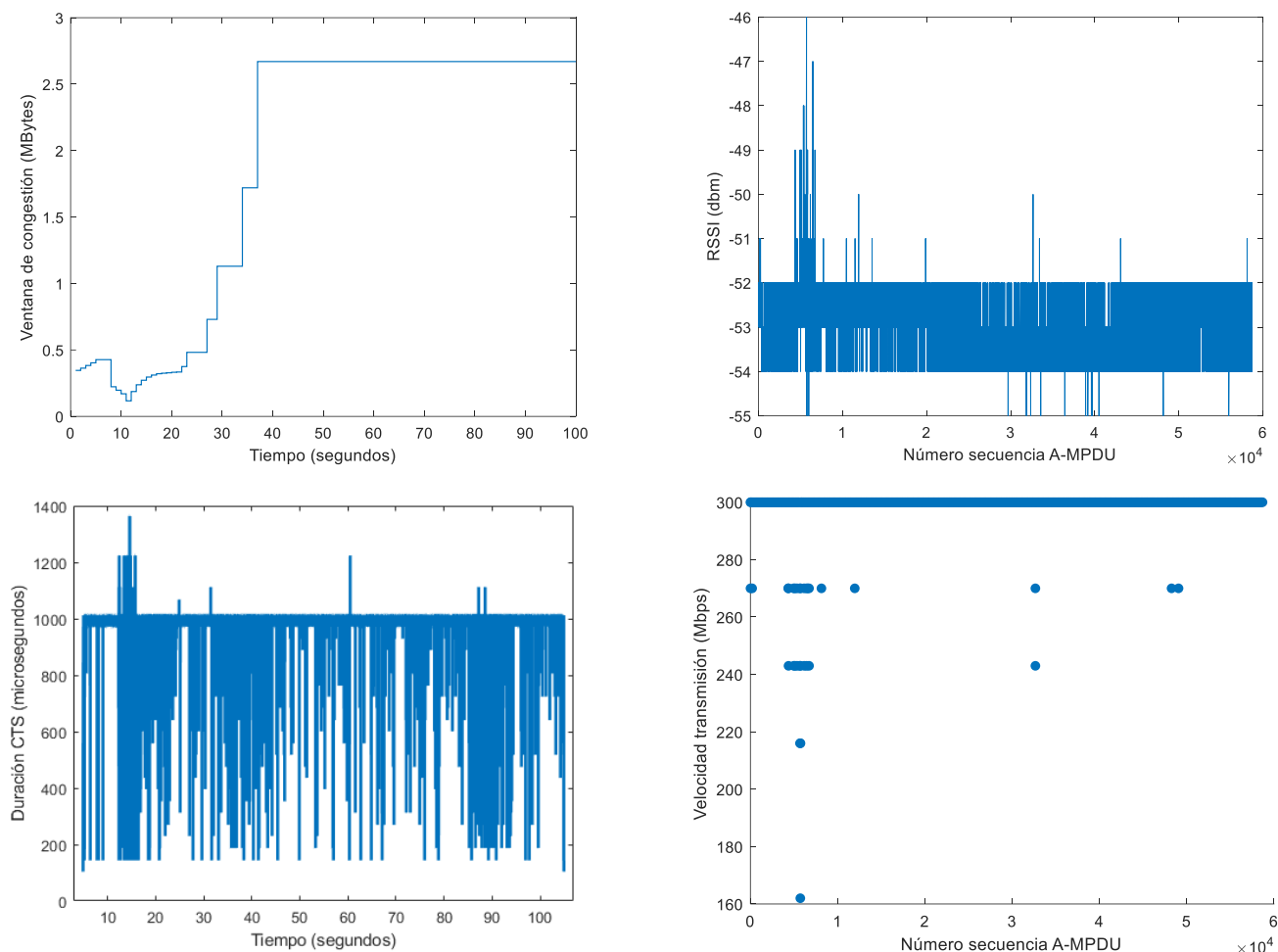
- Canal de 40 MHz, sin tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia sin tráfico interferente empleando el algoritmo de control de congestión Cubic para un canal de 40 MHz.

Tabla 5.10. Resultados medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	131
Volumen datos transmitidos (GBytes)	1.53
Fast Retransmisión (Triple ACK)	6
Retransmisión por Time-Out	6





Figuras 5.18. Resultados de la medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic

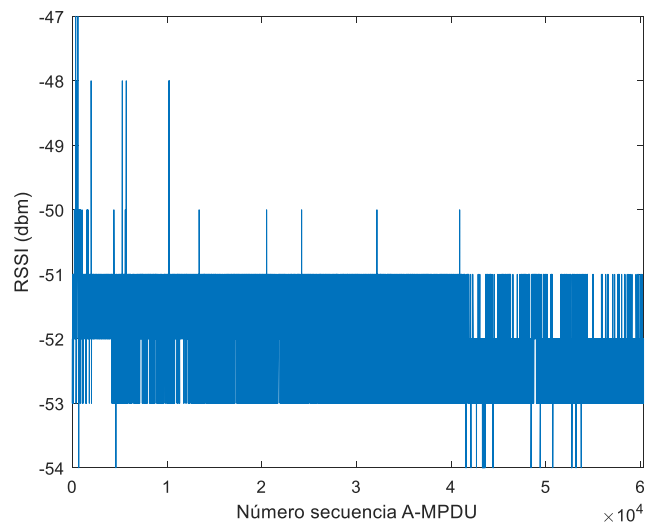
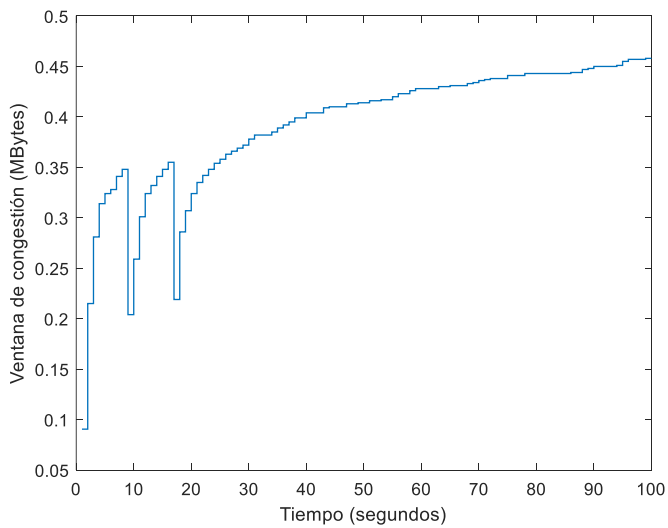
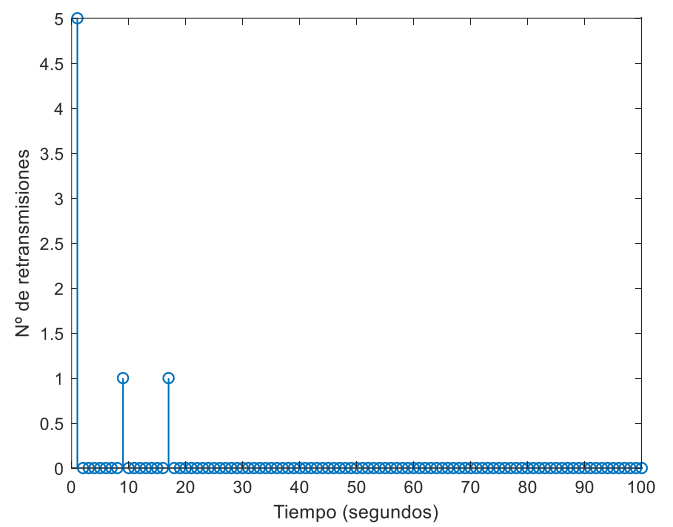
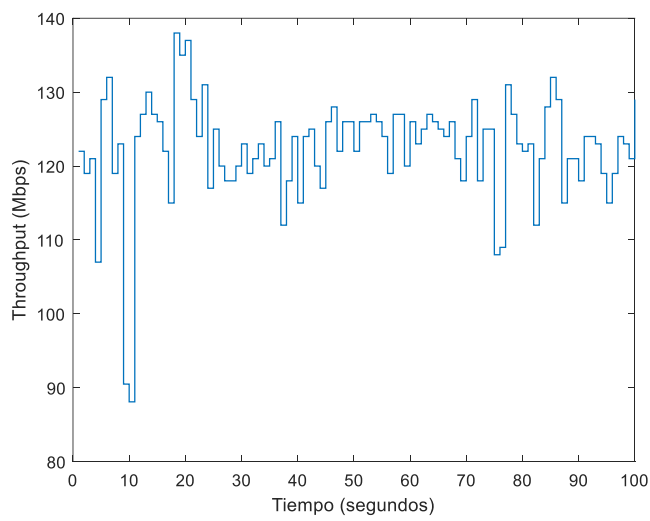
Al utilizar un canal de 40 MHz, las tasas de transmisión ascienden a los 300 Mbps o valores muy cercanos y con ello el throughput, a 131 Mbps. Pero también aparecen retransmisiones sin necesidad de tráfico interferente. Otra diferencia con el canal de 20 MHz es que le otorga el receptor al transmisor la mitad de tiempo para transmitir, de 2000 a 1000 microsegundos, lo cuál es compensado gracias a las altas tasas de transmisión.

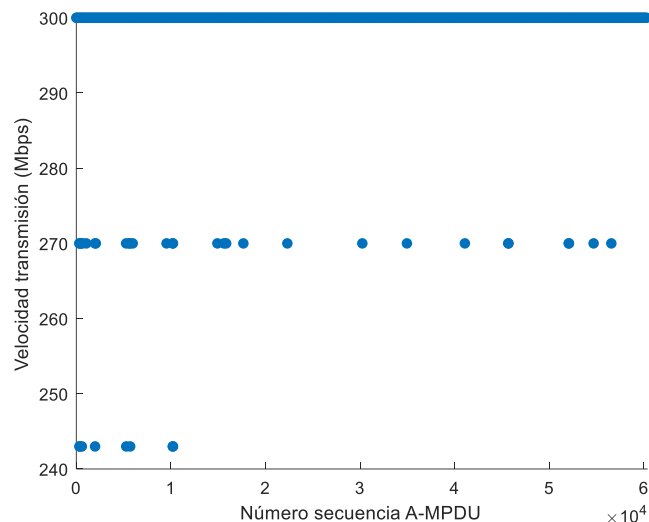
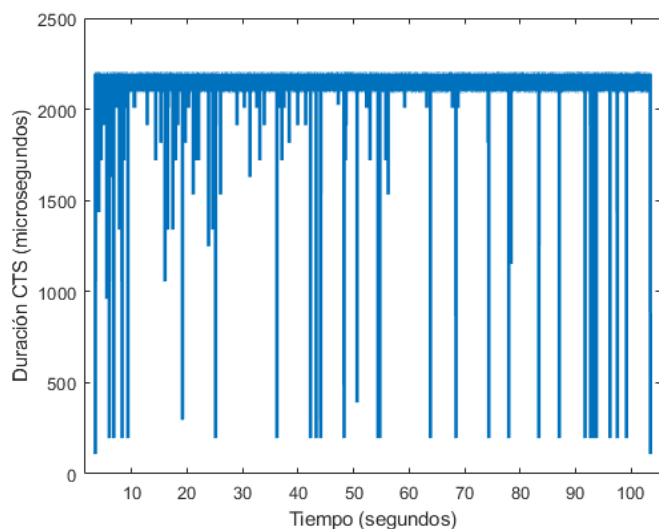
- Canal de 40 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia sin tráfico interferente empleando el algoritmo de control de congestión New Reno para un canal de 40 MHz.

Tabla 5.11. Resultados medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	122
Volumen datos transmitidos (GBytes)	1.42
Fast Retransmisión (Triple ACK)	4
Retransmisión por Time-Out	3





Figuras 5.19. Resultados de la medida del escenario de menor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno

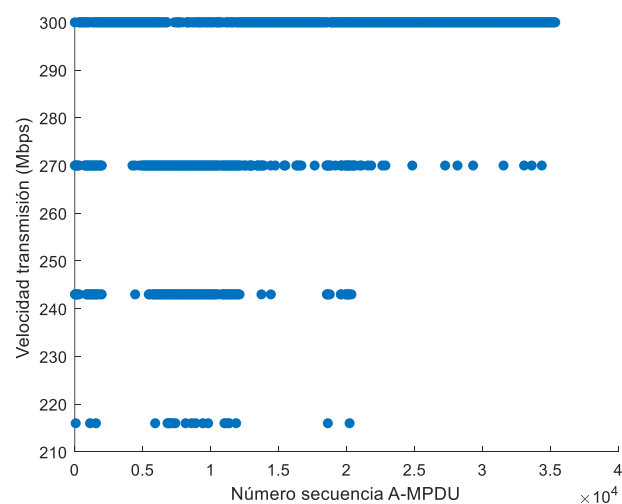
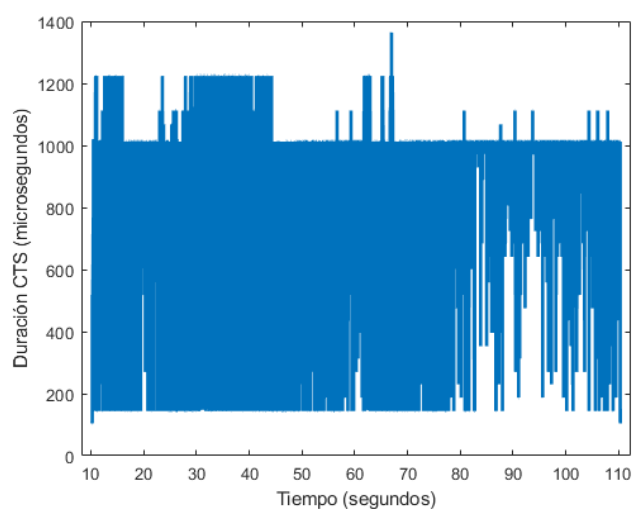
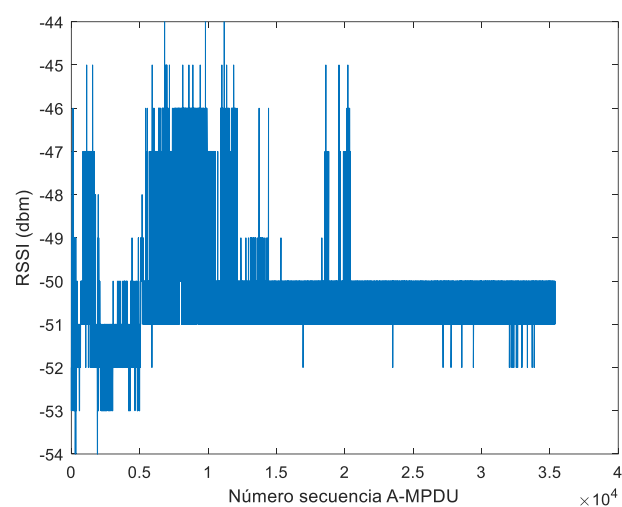
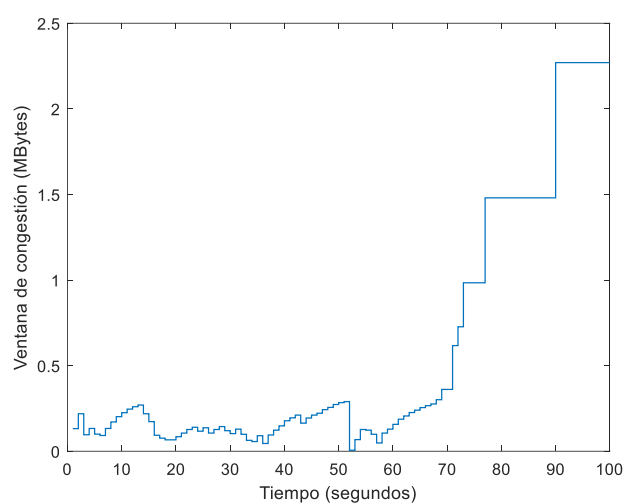
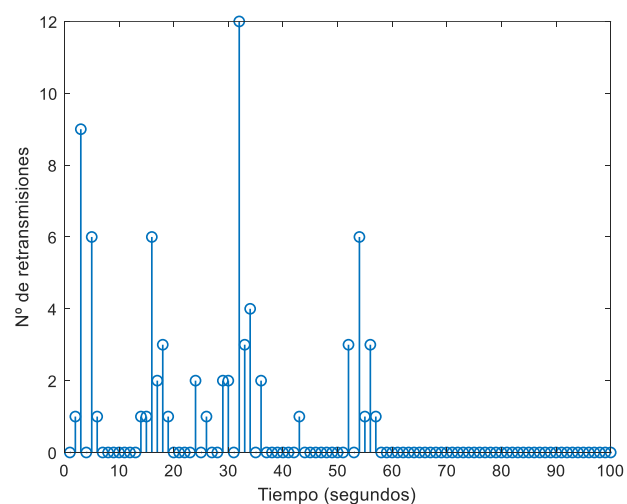
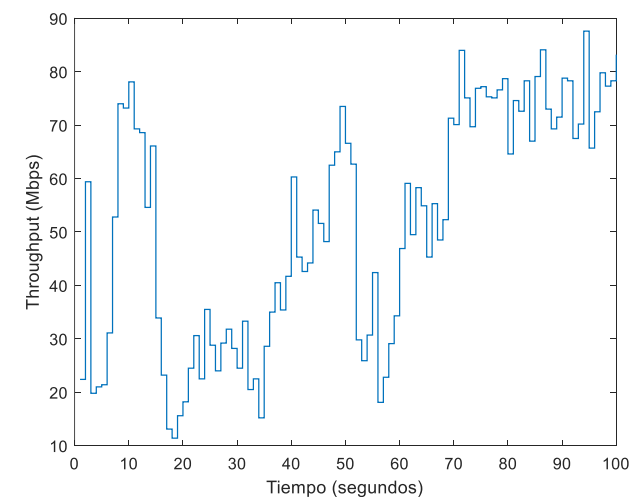
Al igual que en la medida anterior, la tasa de transmisión es prácticamente de 300 Mbps y con un throughput de 122 Mbps. En esta medida se puede observar claramente como las retransmisiones producen reducciones en la ventana de congestión.

- Canal de 40 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia con tráfico interferente empleando el algoritmo de control de congestión Cubic para un canal de 40 MHz.

Tabla 5.12. Resultados medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	51.6
Volumen datos transmitidos (MBytes)	615
Fast Retransmisión (Triple ACK)	25
Retransmisión por Time-Out	49



Figuras 5.20. Resultados de la medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando Cubic

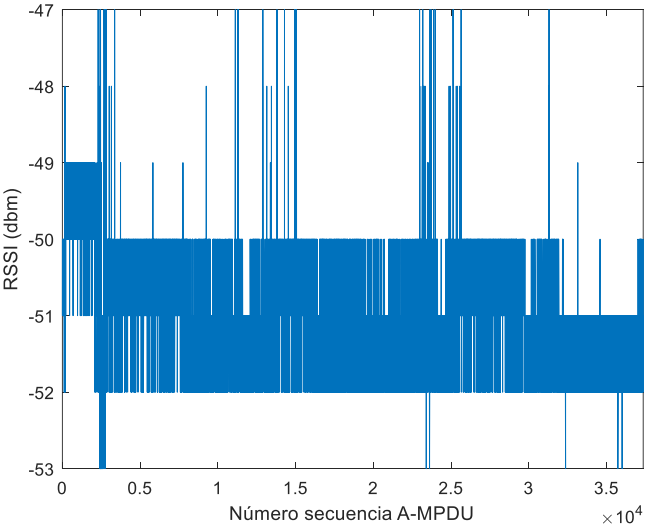
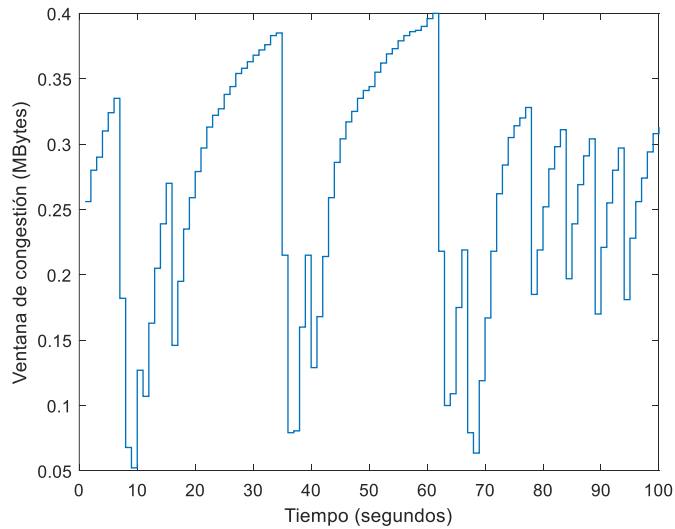
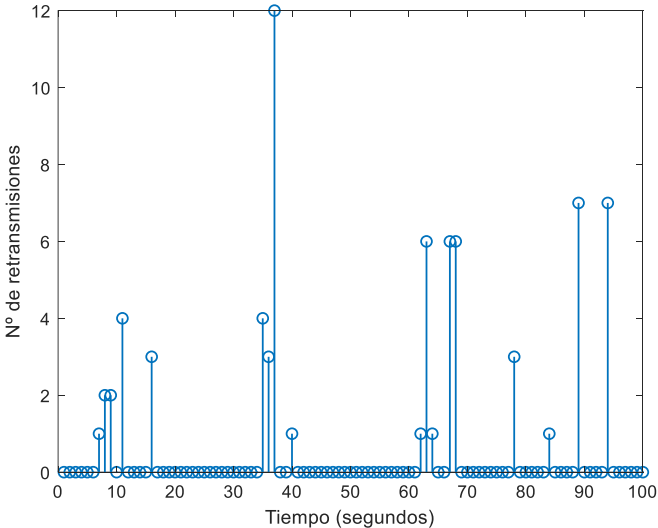
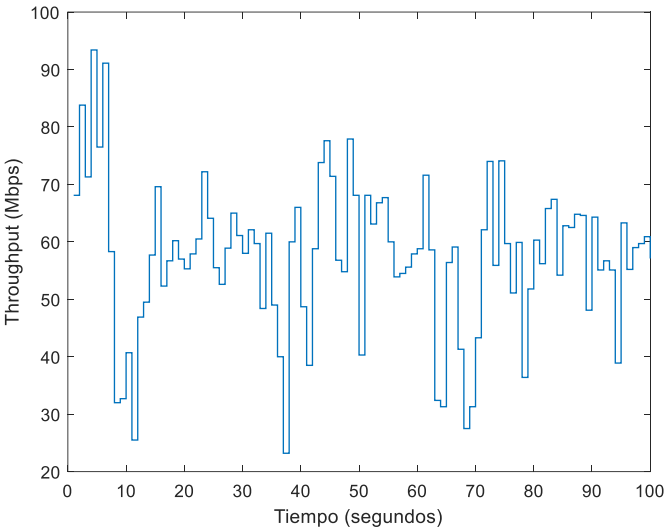
En un canal de 40 MHz con tráfico interferente, el throughput se ha reducido a 51.6 Mbps, ha aumentado el número de retransmisiones TCP y si bien que la tasa de transmisión más común es 300 Mbps, ya aparecen muchas más ráfagas A-MPDU a velocidades de transmisión menores.

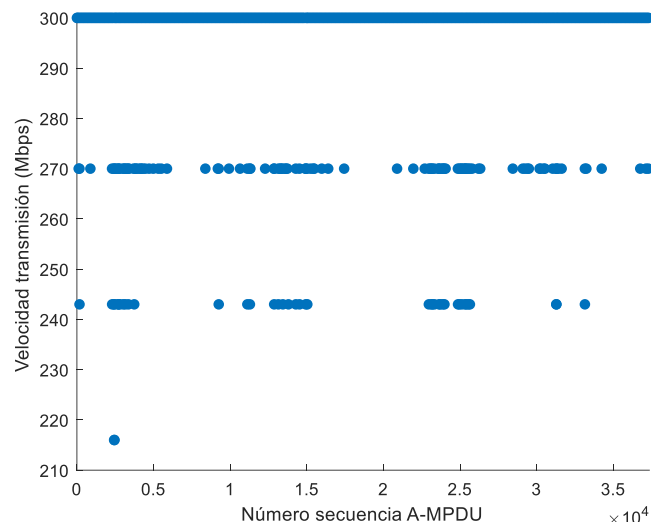
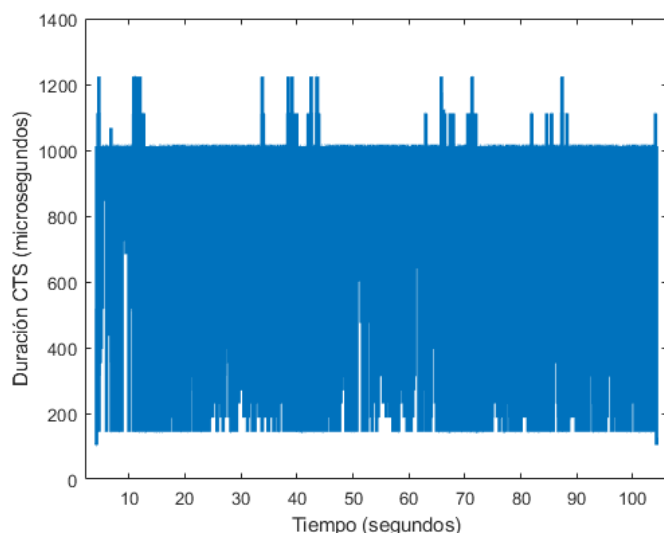
- Canal de 40 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de menor distancia con tráfico interferente empleando el algoritmo de control de congestión New Reno para un canal de 40 MHz.

Tabla 5.13. Resultados medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	57.4
Volumen datos transmitidos (MBytes)	684
Fast Retransmisión (Triple ACK)	22
Retransmisión por Time-Out	48





Figuras 5.21. Resultados de la medida del escenario de menor distancia canal de 40 MHz con tráfico interferente utilizando New Reno

En esta medida el throughput es de 57.4 Mbps y las tasas de transmisión siguen estando cercanas a los 300 Mbps. También se puede observar tanto en esta medida como en la anterior, como el tiempo otorgado al transmisor para enviar información varía entre 200 y 1000 microsegundos.

5.3.2. Escenario de distancia media

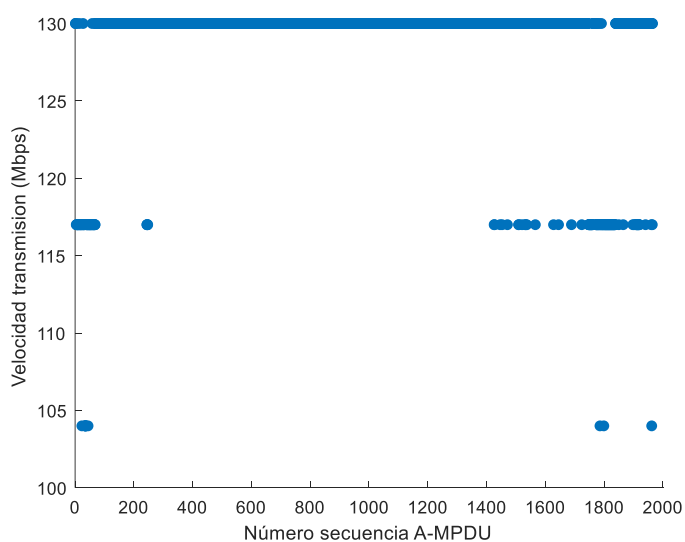
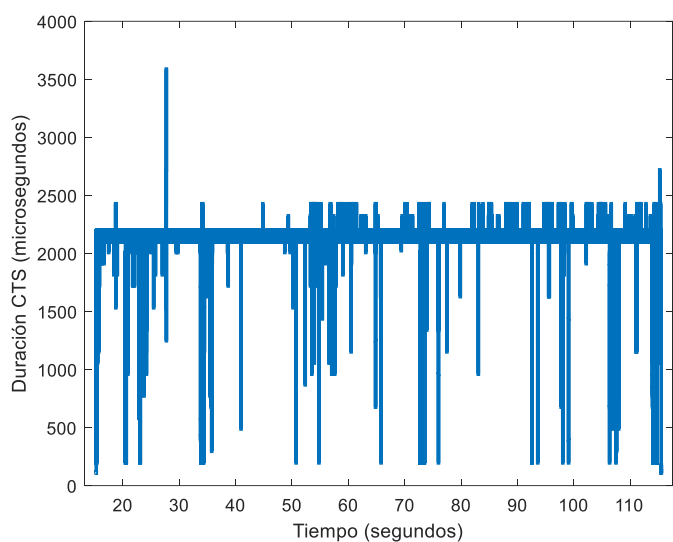
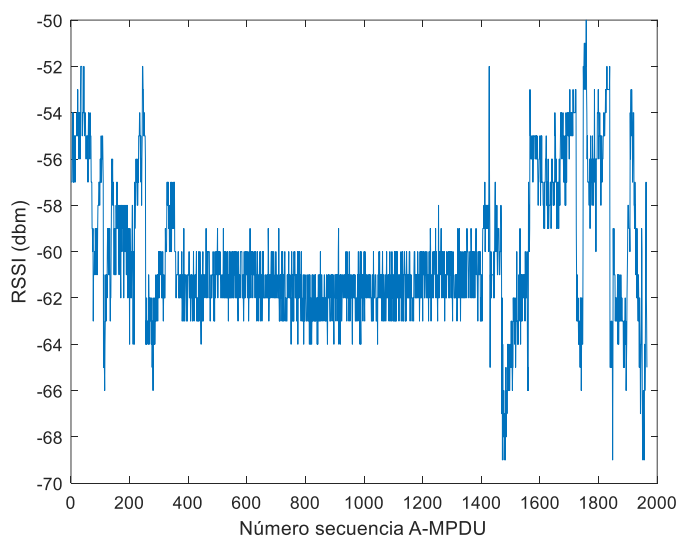
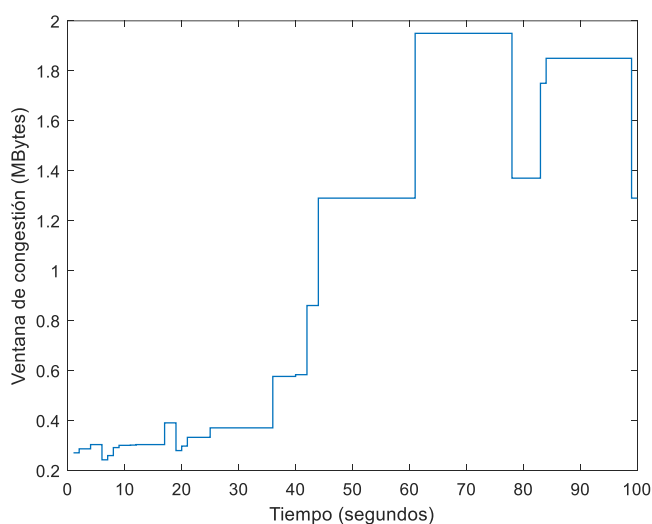
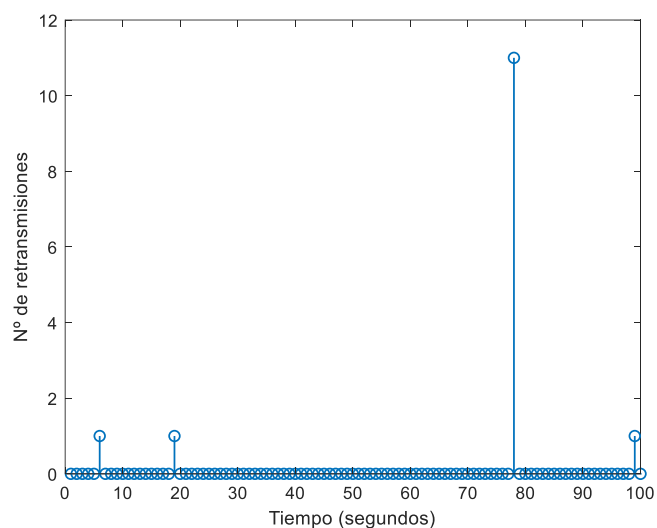
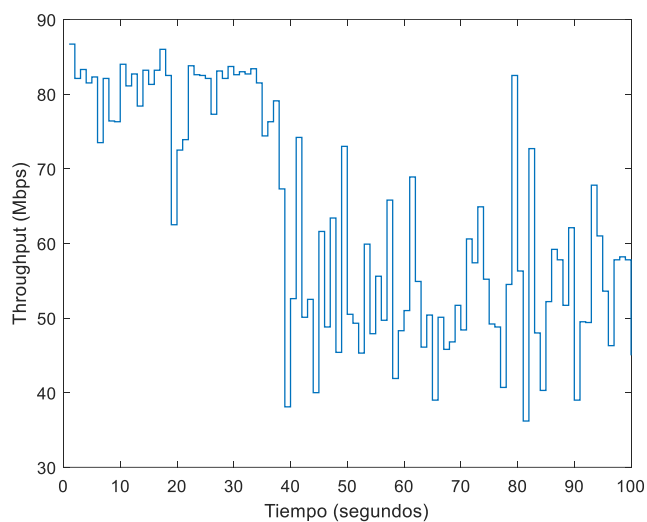
En este escenario tanto transmisor y receptor estarán alejados a una distancia media. Para ver más información sobre el escenario de distancia media consultar el apartado 3.2.2.

- Canal de 20 MHz, sin tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media sin tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 20 MHz.

Tabla 5.14. Resultados medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	63.5
Volumen datos transmitidos (MBytes)	756
Fast Retransmisión (Triple ACK)	4
Retransmisión por Time-Out	10



Figuras 5.22. Resultados de la medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando Cubic

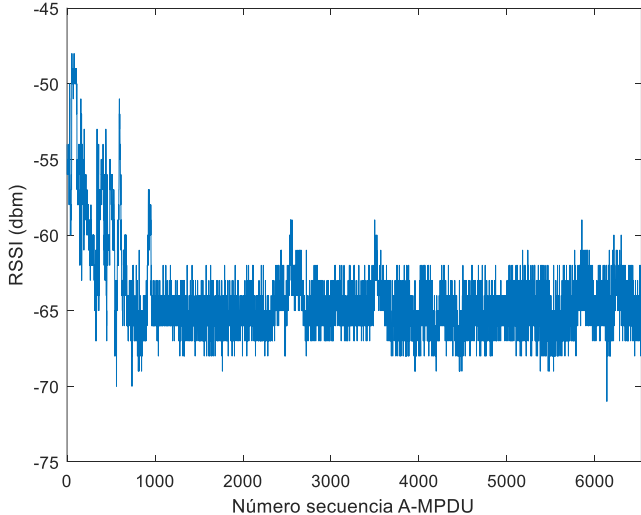
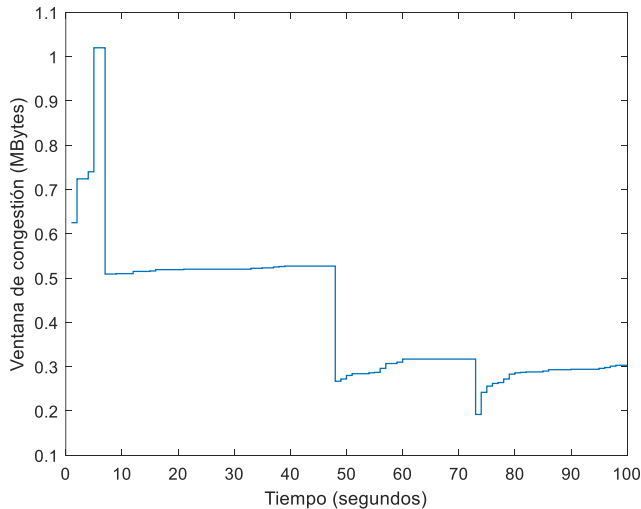
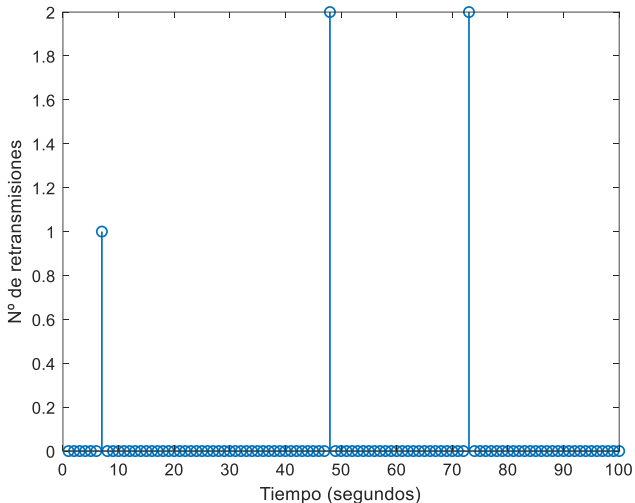
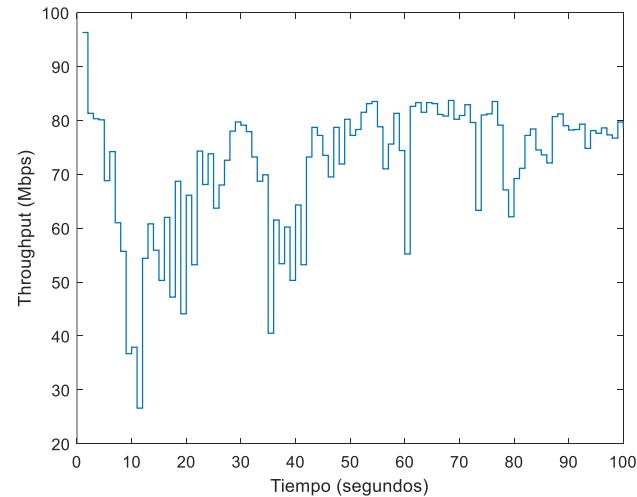
En esta medida, al ser el escenario de distancia media, el throughput se ha reducido a 63.5 Mbps y se producen retransmisiones TCP sin presencia de tráfico interferente en un canal de 20 MHz, pero la velocidad de transmisión aún está alrededor de los 130 Mbps.

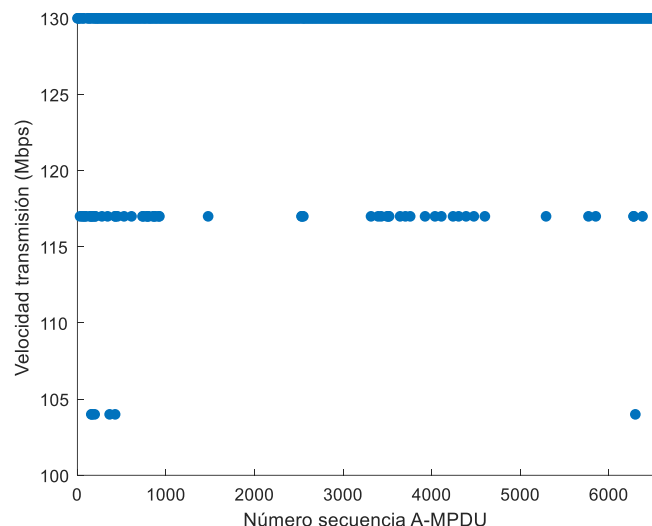
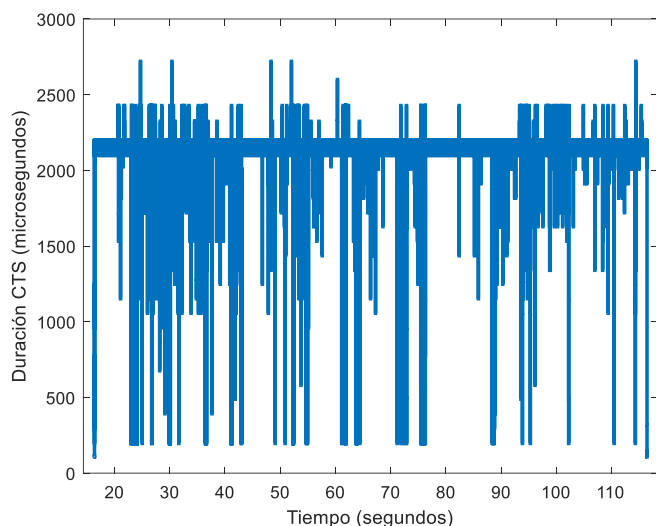
- Canal de 20 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media sin tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 20 MHz.

Tabla 5.15. Resultados medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	71.5
Volumen datos transmitidos (MBytes)	852
Fast Retransmisión (Triple ACK)	3
Retransmisión por Time-Out	2





Figuras 5.23. Resultados de la medida del escenario distancia media canal de 20 MHz sin tráfico interferente utilizando New Reno

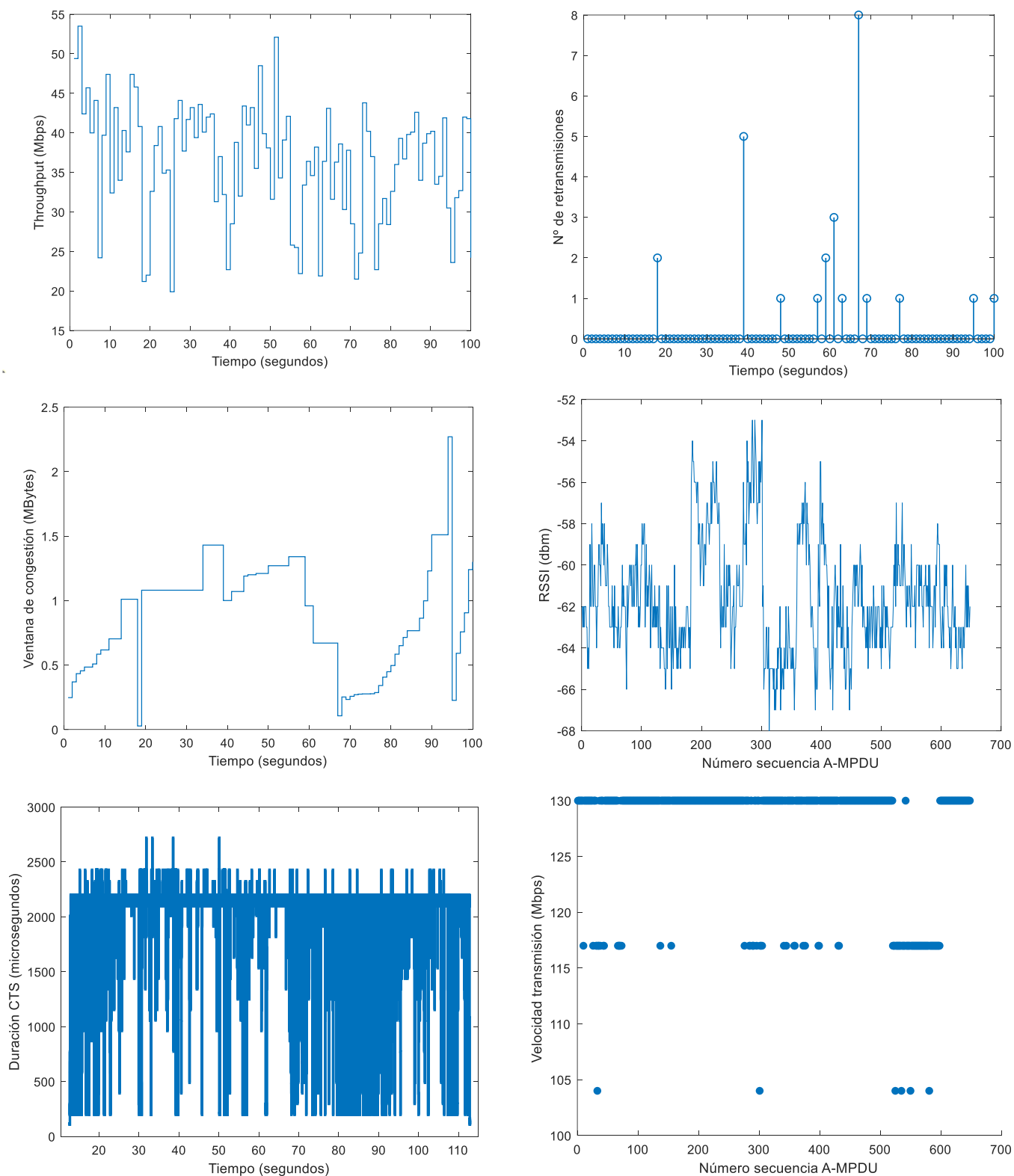
Al igual que la medida anterior, el throughput se ha reducido respecto al mismo caso pero para un escenario de menor distancia. También se ve que la velocidad de transmisión común es 130 Mbps pero ya empieza a aparecer ráfagas A-MPDU a velocidades inferiores.

- Canal de 20 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media con tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 20 MHz.

Tabla 5.16. Resultados medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	36.4
Volumen datos transmitidos (MBytes)	433
Fast Retransmisión (Triple ACK)	6
Retransmisión por Time-Out	21



Figuras 5.24. Resultados de la medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando Cubic

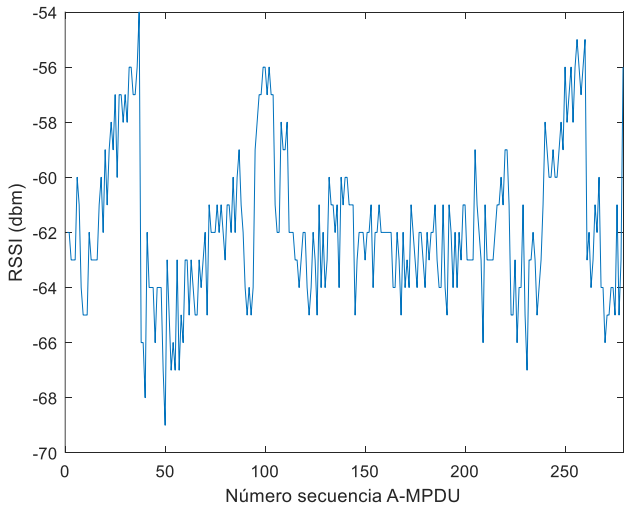
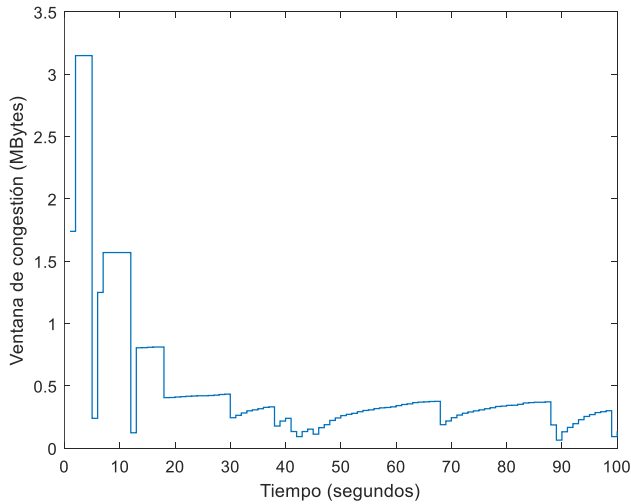
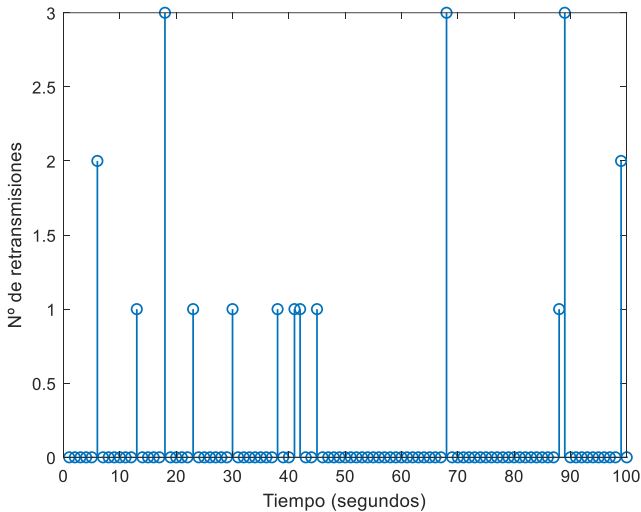
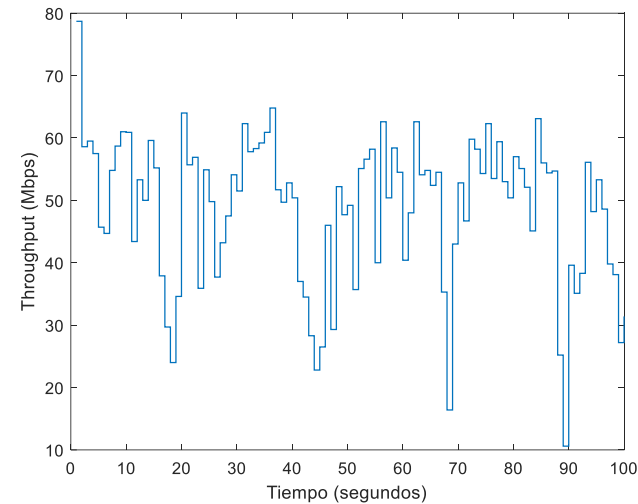
Al introducir tráfico interferente, el throughput se reduce a 36.4 Mbps y el número de retransmisiones aumenta. Las velocidades de transmisión, como ya se ha comentado anteriormente, prácticamente no ha cambiado estando en torno a los 130 Mbps.

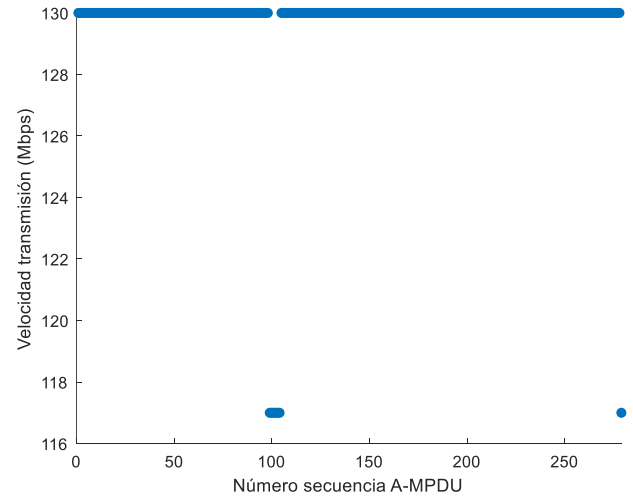
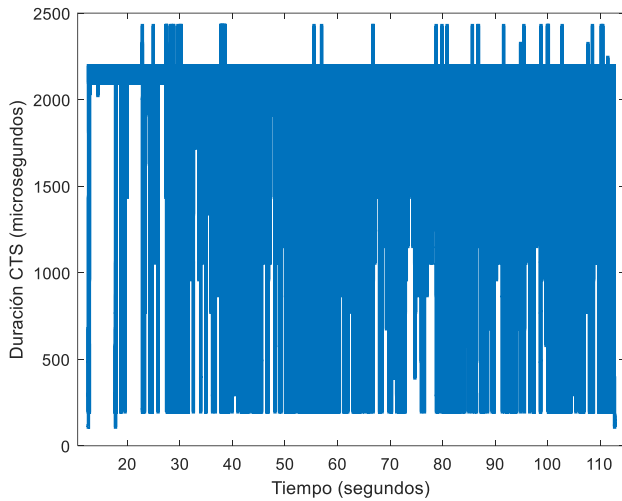
- Canal de 20 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media con tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 20 MHz.

Tabla 5.17. Resultados medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	48.7
Volumen datos transmitidos (MBytes)	581
Fast Retransmisión (Triple ACK)	10
Retransmisión por Time-Out	11





Figuras 5.25. Resultados de la medida del escenario distancia media canal de 20 MHz con tráfico interferente utilizando New Reno

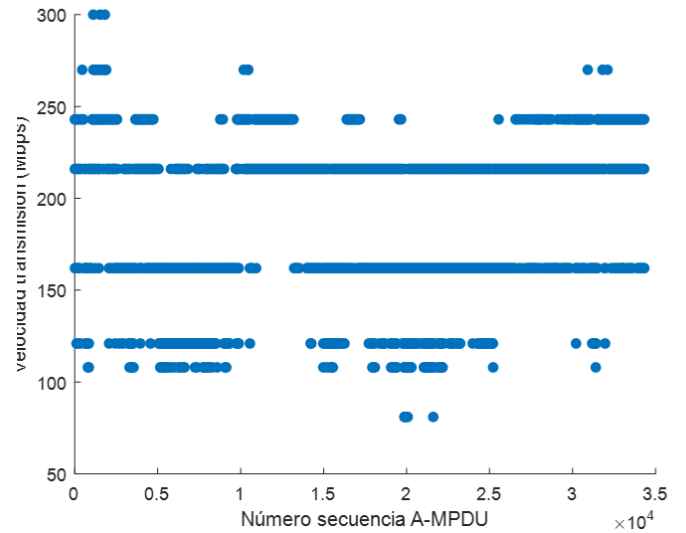
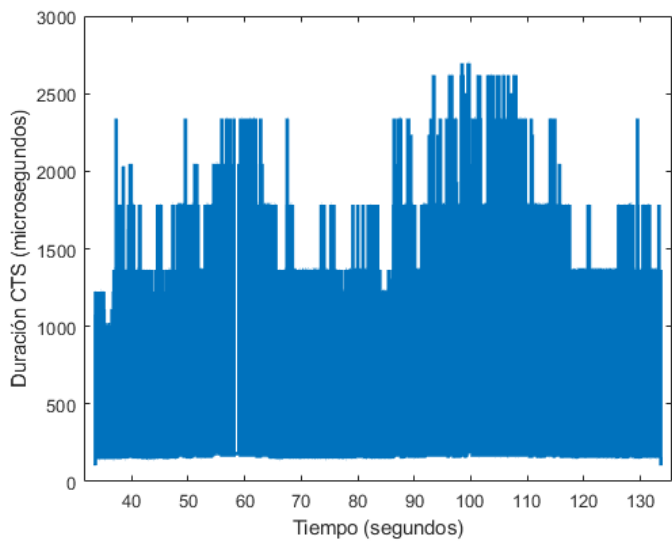
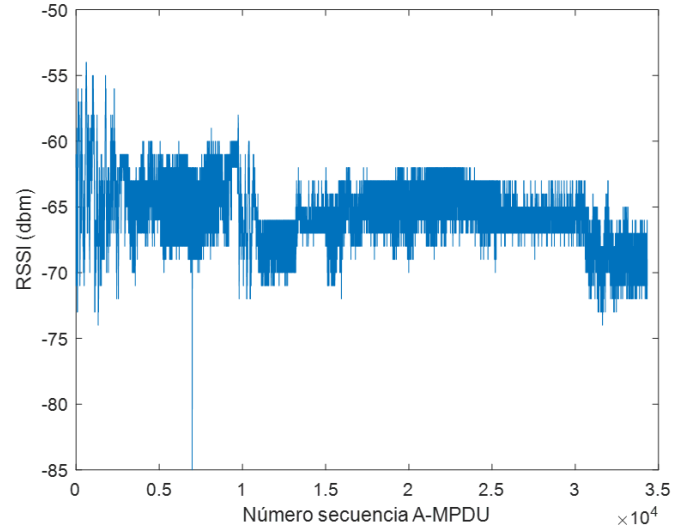
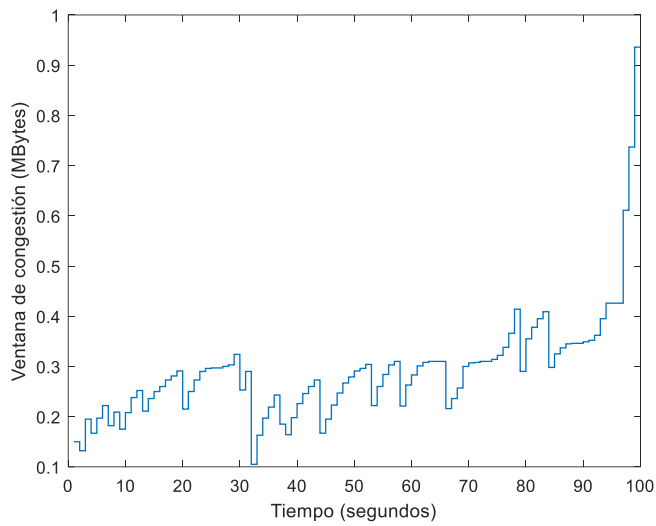
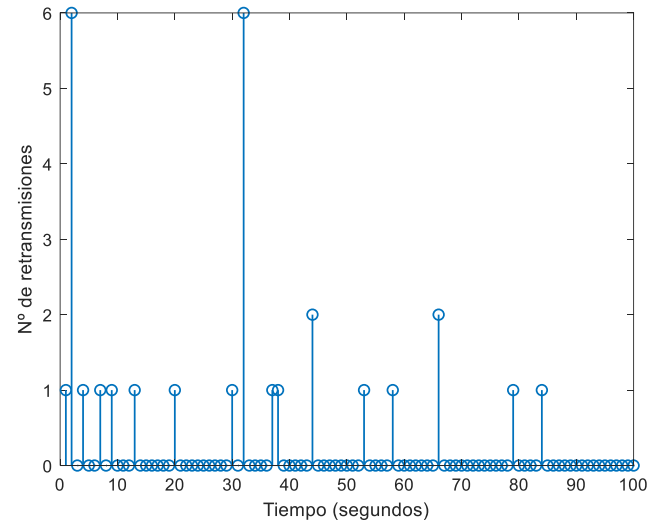
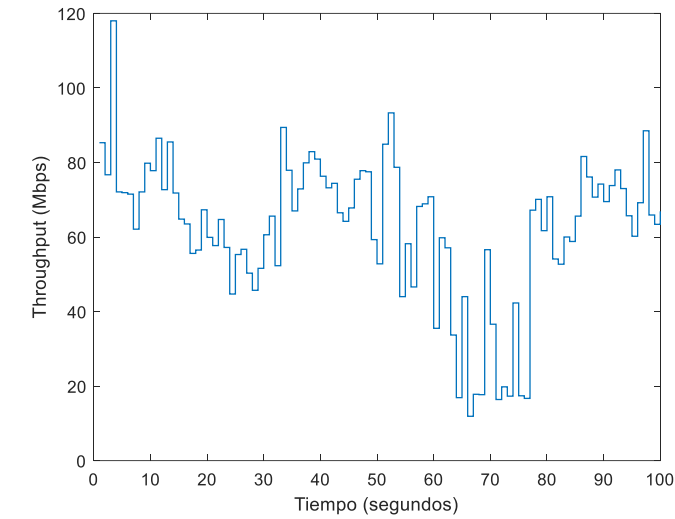
En esta medida ocurre lo mismo que en la anterior. La presencia de tráfico interferente produce una reducción del throughput y un aumento de las retransmisiones, sin cambiar la velocidad de transmisión.

- Canal de 40 MHz, sin tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media sin tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 40 MHz.

Tabla 5.18. Resultados medida del Escenario distancia media Canal de 40 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	62.2
Volumen datos transmitidos (MBytes)	741
Fast Retransmisión (Triple ACK)	21
Retransmisión por Time-Out	8



Figuras 5.26. Resultados de la medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando Cubic

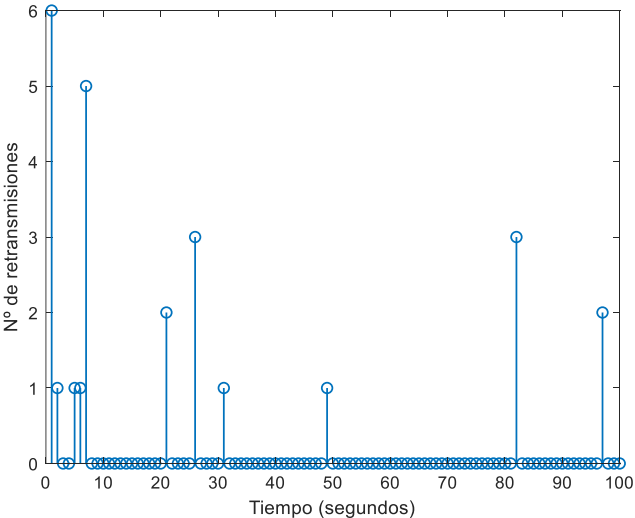
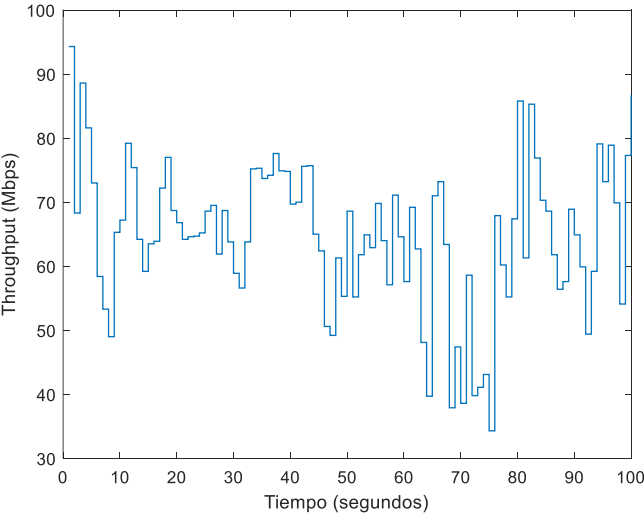
Al utilizar un canal de 40 MHz sin tráfico interferente en el escenario de distancia media, en este caso el throughput se ha mantenido en unos valores similares a los de un canal de 20 MHz. A diferencia del escenario de menor distancia, aquí las velocidades de transmisión para un canal de 40 MHz empiezan a variar claramente, habiendo incluso unas pocas A-MPDUs por debajo de los 100 Mbps, siendo las velocidades de transmisión más comunes 243, 216, 162 y 120 Mbps.

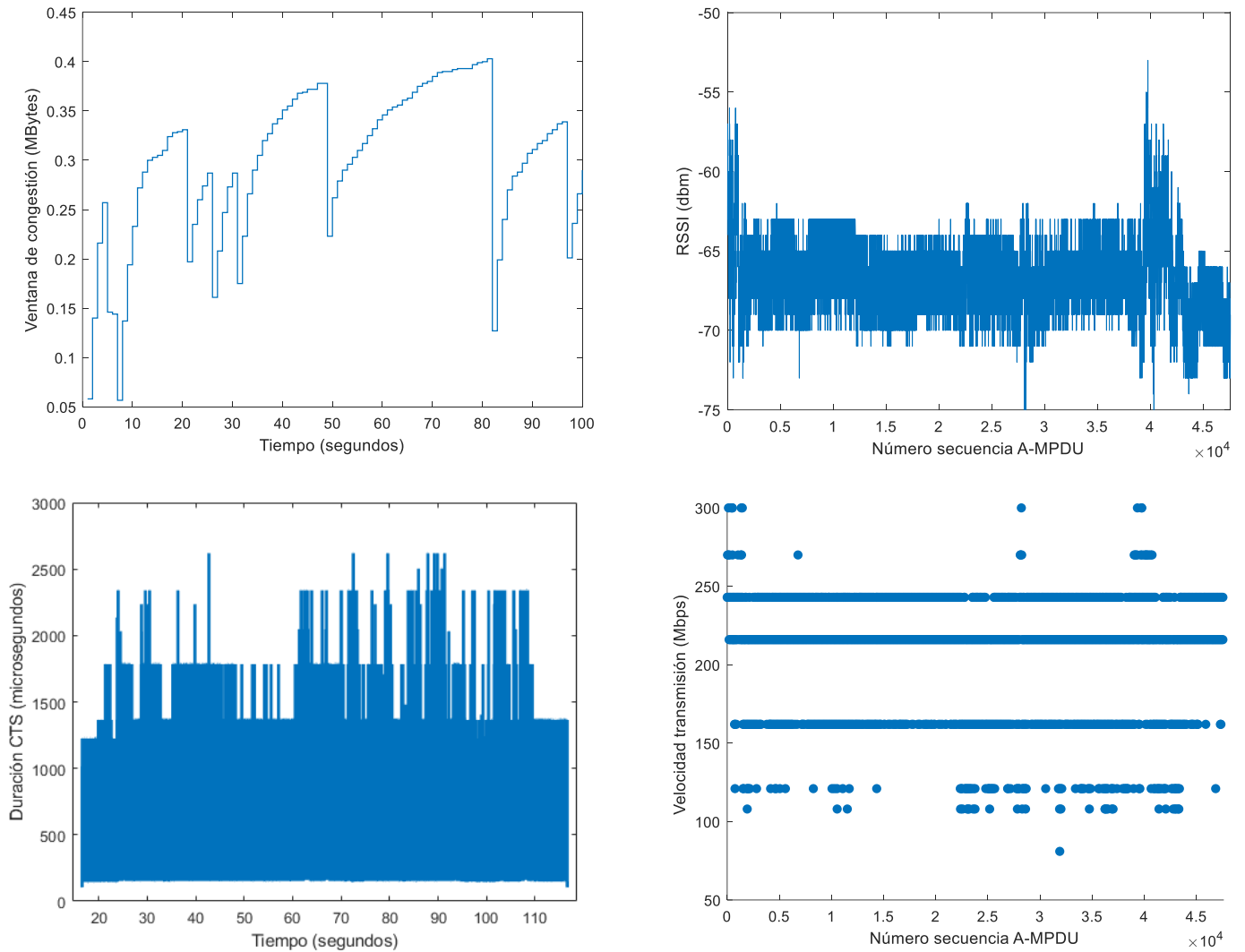
- Canal de 40 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media sin tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 40 MHz.

Tabla 5.19. Resultados medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	64.8
Volumen datos transmitidos (MBytes)	773
Fast Retransmisión (Triple ACK)	17
Retransmisión por Time-Out	9





Figuras 5.27. Resultados de la medida del escenario distancia media canal de 40 MHz sin tráfico interferente utilizando New Reno

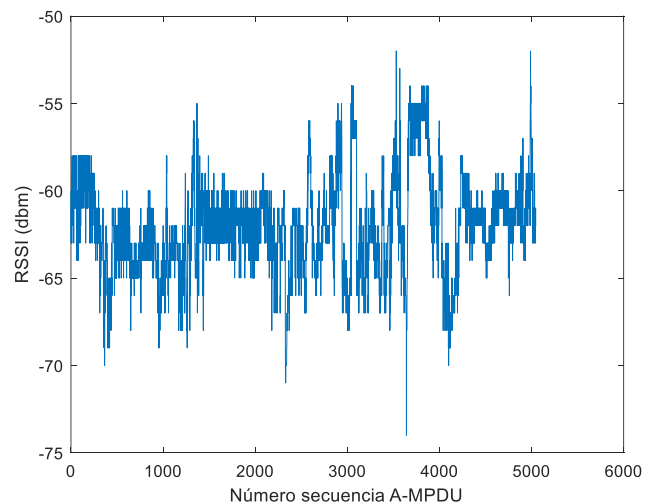
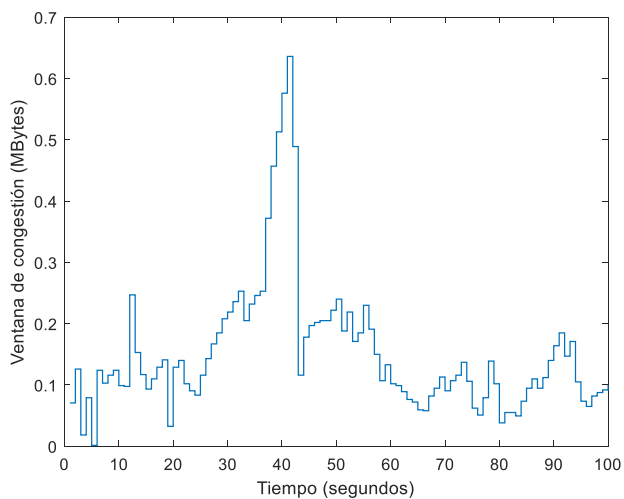
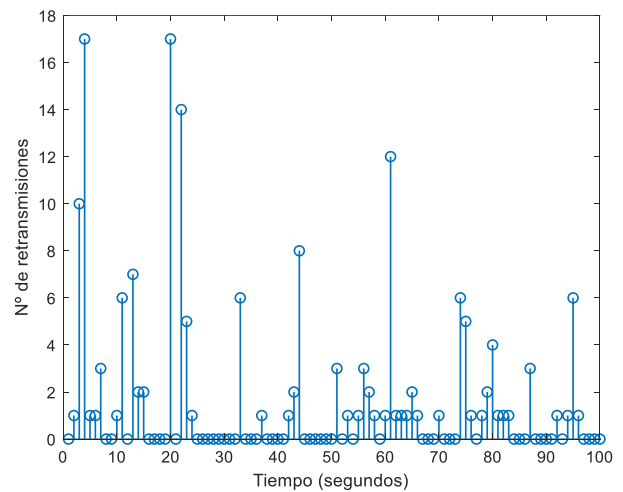
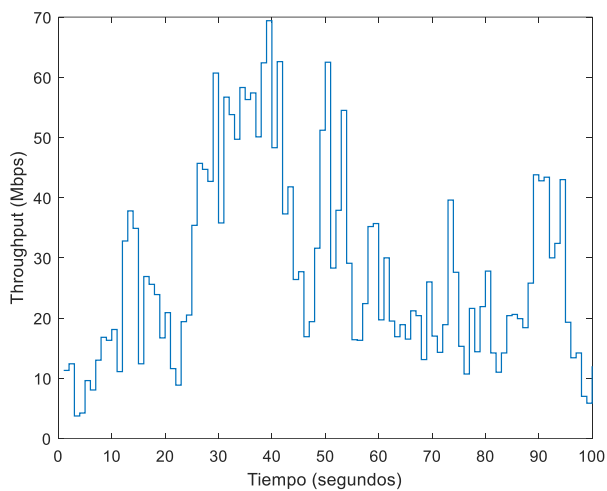
Al igual que antes, el throughput es similar a un canal de 20 MHz, aumentan las retransmisiones y la velocidad de transmisión ya no es estable. Aquí de nuevo, se vuelve a reflejar en la gráfica de la evolución de la ventana de congestión el crecimiento gradual debido al uso de New Reno. Otro dato característico del escenario de distancia media es que la potencia de recepción de señal se sitúa entre los -65 y los -69 dBm, mientras que en el escenario de menor distancia la potencia se situaba en torno a los -55 dBm.

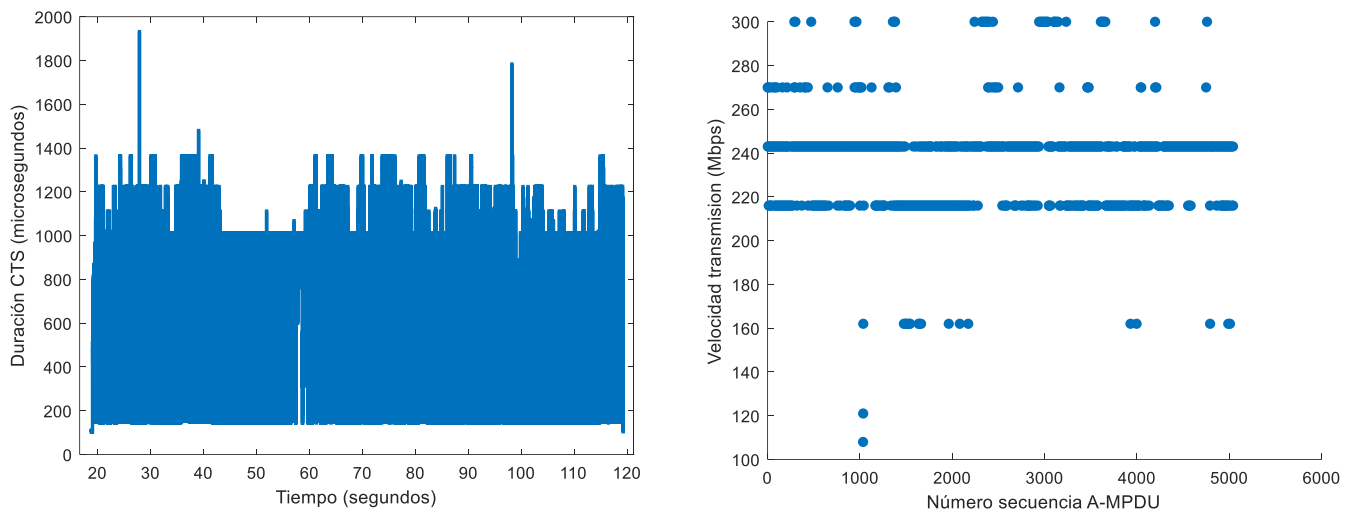
- Canal de 40 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media con tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 40 MHz.

Tabla 5.20. Resultados medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	27.8
Volumen datos transmitidos (MBytes)	331
Fast Retransmisión (Triple ACK)	29
Retransmisión por Time-Out	142





Figuras 5.28. Resultados de la medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando Cubic

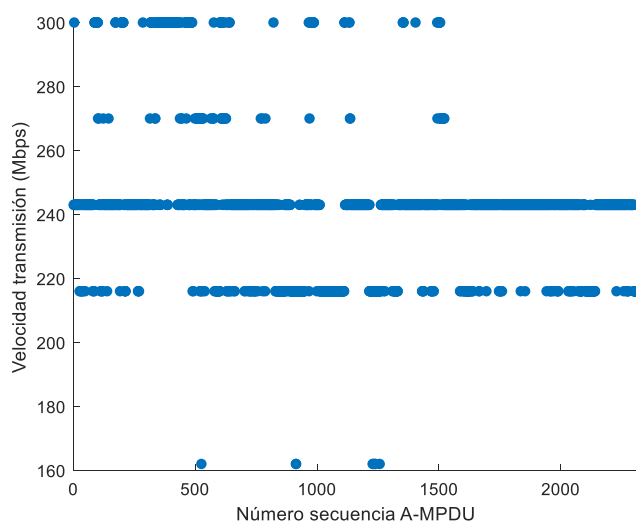
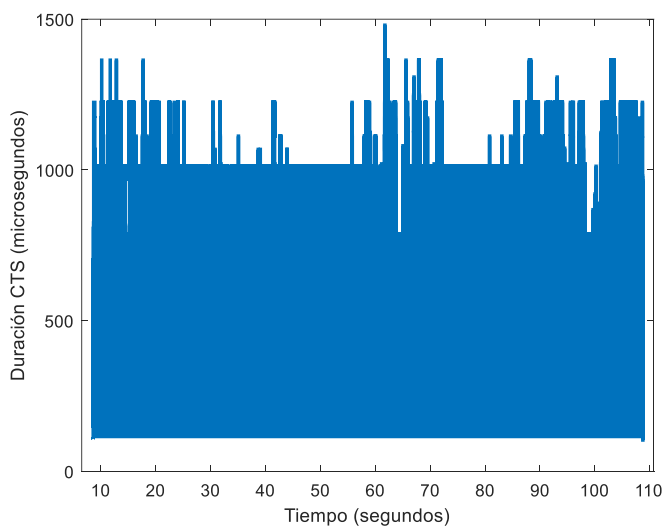
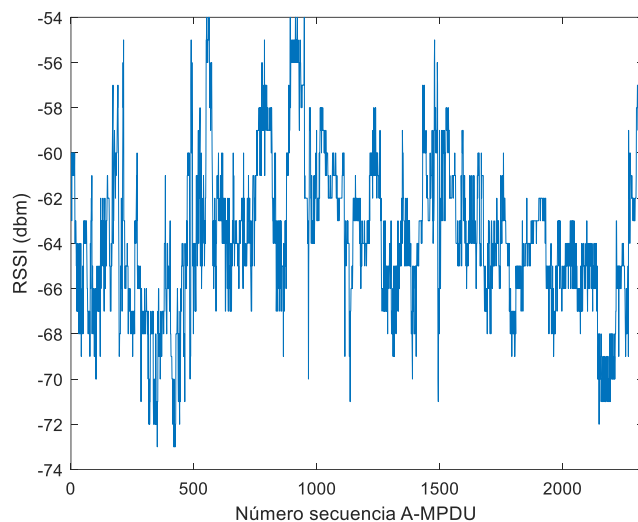
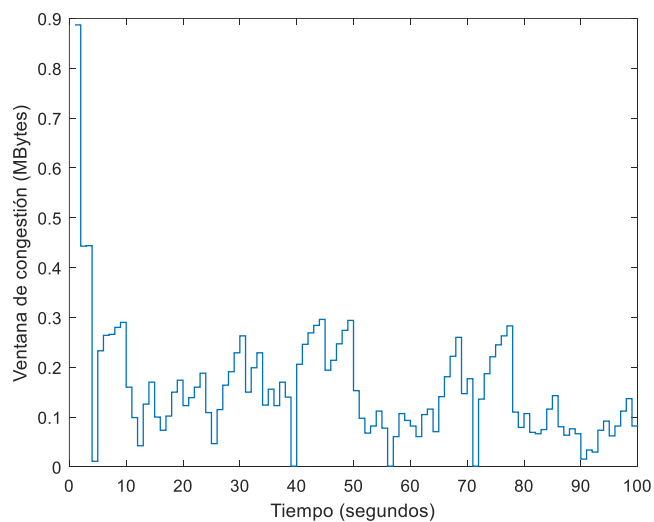
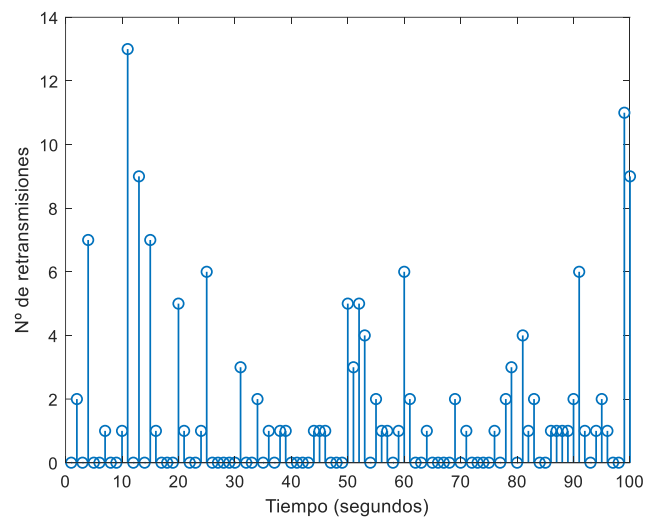
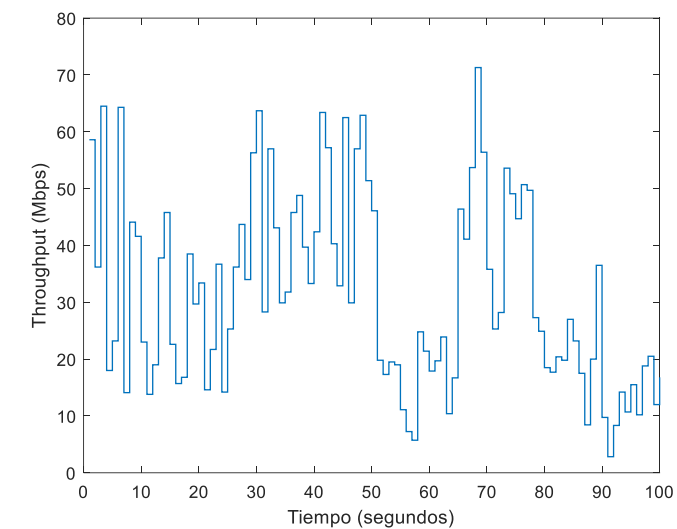
Al añadir tráfico interferente, se reduce el throughput y aumenta el número de retransmisiones. Los tiempos otorgados al transmisor para enviar información a consecuencia de ello se reducen.

- Canal de 40 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de distancia media con tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 40 MHz.

Tabla 5.21. Resultados medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	31.5
Volumen datos transmitidos (MBytes)	376
Fast Retransmisión (Triple ACK)	29
Retransmisión por Time-Out	120



Figuras 5.29. Resultados de la medida del escenario distancia media canal de 40 MHz con tráfico interferente utilizando New Reno

Utilizar New Reno no cambia nada, se sigue produciendo una reducción del throughput y un aumento de las retransmisiones TCP debido a la presencia de tráfico interferente.

5.3.3. Escenario de mayor distancia

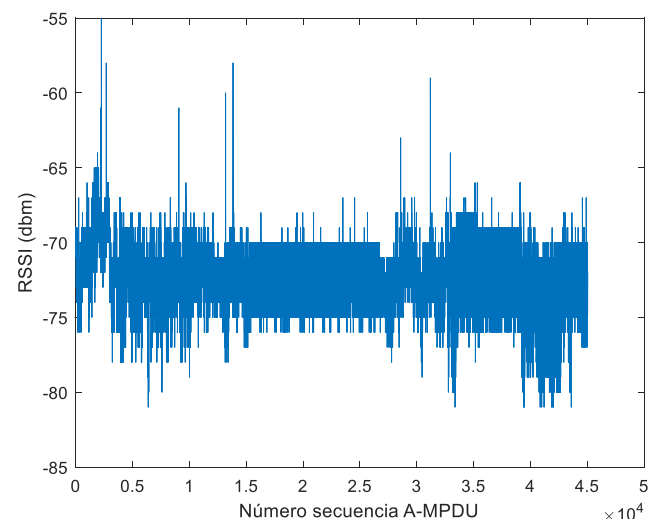
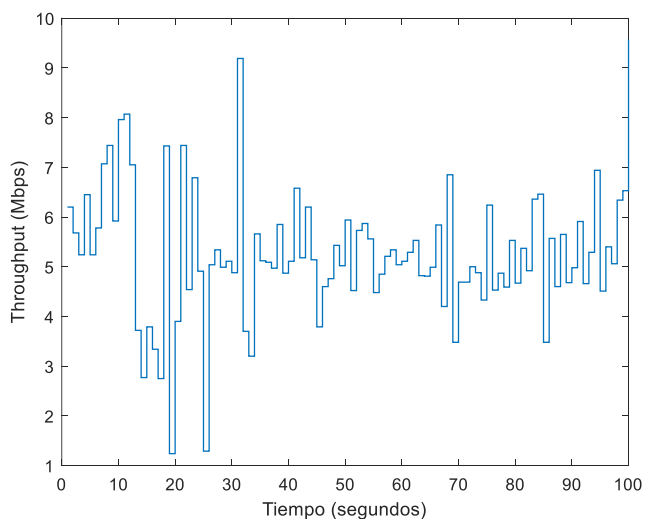
En este escenario tanto transmisor y receptor estarán lo más alejados posible. Para ver más información sobre el escenario de mayor distancia consultar el apartado 3.2.3.

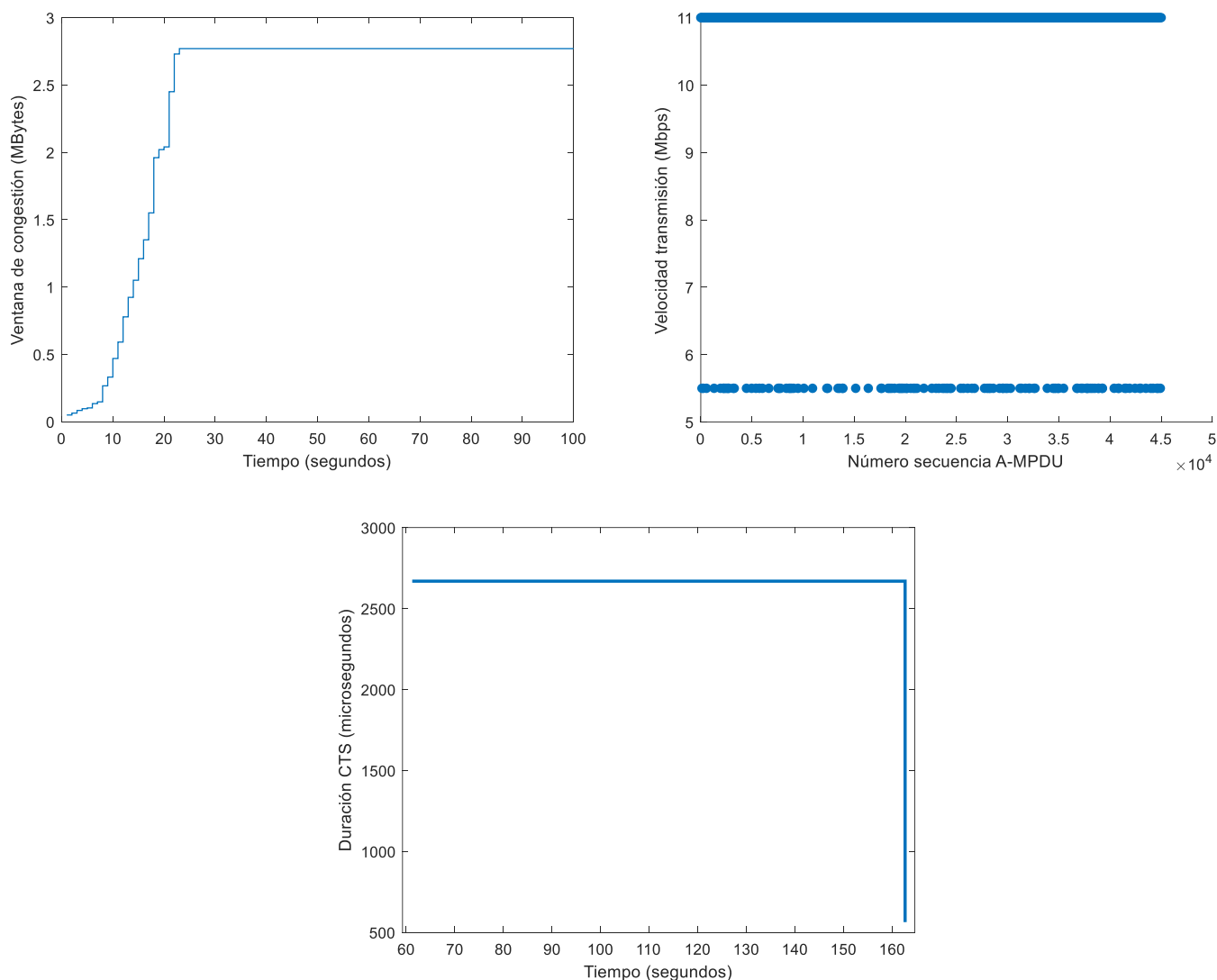
- Canal de 20 MHz, sin tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia sin tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 20 MHz.

Tabla 5.22. Resultados medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	5.27
Volumen datos transmitidos (MBytes)	62.8
Nº Retransmisiones	0





Figuras 5.30. Resultados de la medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando Cubic

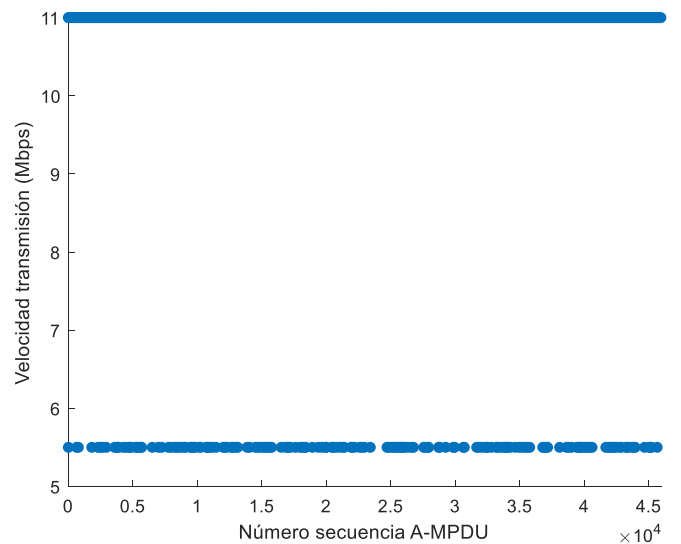
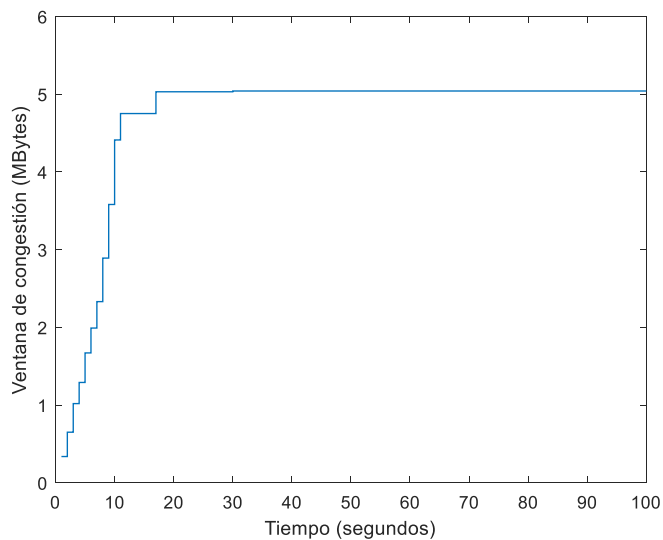
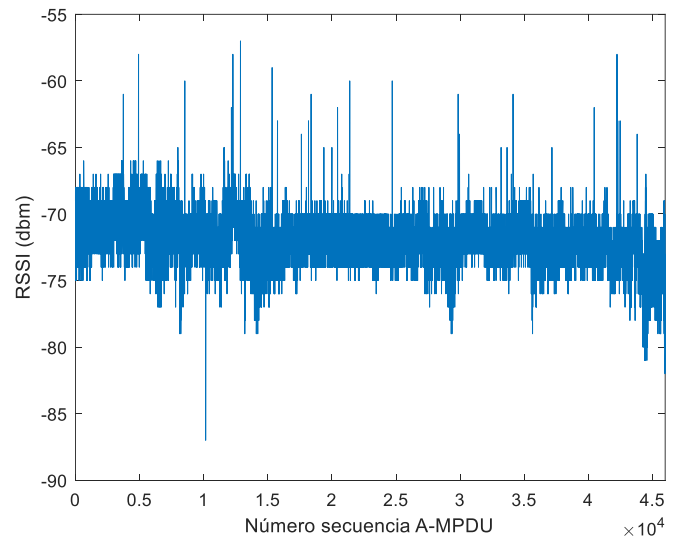
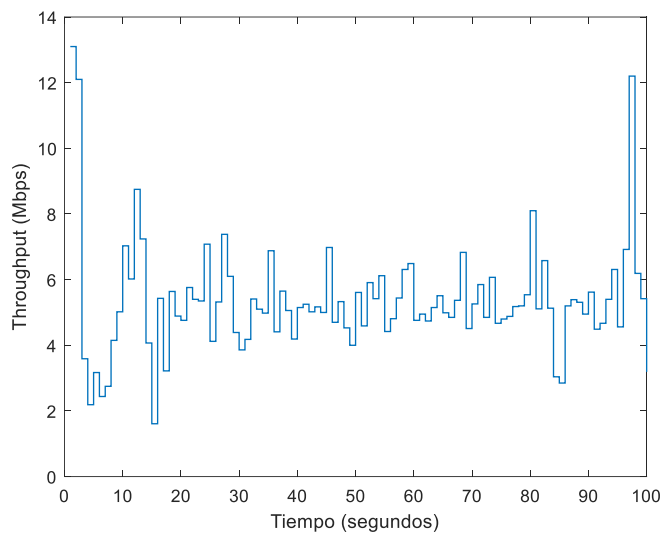
En el escenario de mayor distancia en un canal de 20 MHz no se producen retransmisiones debido al mecanismo de adaptación de velocidad de transmisión de Atheros, que evita que se produzcan retransmisiones TCP reduciendo drásticamente las velocidades de transmisión, siendo 11 y 5.5 Mbps las velocidades utilizadas. Son tan bajas las velocidades de transmisión que ya ni siquiera se emplea el estándar 802.11n, sino el estándar 802.11b. La potencia de recepción en este escenario se sitúa entre -70 y -75 dBm. También se produce una caída del throughput a 5.27 Mbps. La duración que se otorga al transmisor para poder enviar información se mantiene estable debido a las bajas velocidades de transmisión.

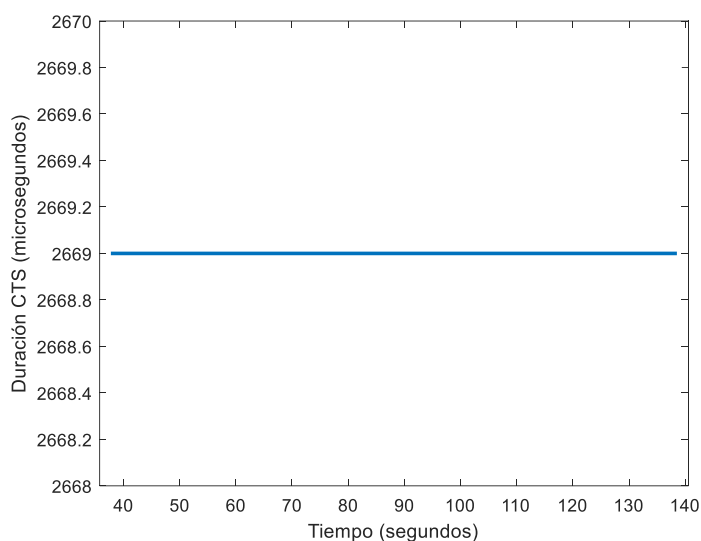
- Canal de 20 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia sin tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 20 MHz.

Tabla 5.23. Resultados medida del escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	5.35
Volumen datos transmitidos (MBytes)	63.7
Nº Retransmisiones	0





Figuras 5.31. Resultados de la medida del Escenario de mayor distancia canal de 20 MHz sin tráfico interferente utilizando New Reno

El uso de New Reno no influye nada en la medida, siendo prácticamente iguales los resultados obtenidos con New Reno como los obtenidos con Cubic.

- Canal de 20 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia con tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 20 MHz.

Tabla 5.24. Resultados medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	1.17
Volumen datos transmitidos (MBytes)	14
Nº Retransmisiones	0

En la siguiente Figura 5.32, se presenta una peculiaridad. Aproximadamente en el segundo 55, aumenta drásticamente el throughput, pero los segundos siguientes indica un throughput de valor nulo o cercano a este. Este incremento del throughput es posible debido a la conexión sin retransmisiones a causa de las bajas velocidades de transmisión, pero debido a la distancia, el servidor indica un ancho de banda inferior, sobre todo si se mira los resultados de la parte servidor. En esa parte, no aparece indicado ese pico, sino que el valor puntual de throughput se divide entre los segundos donde el throughput el transmisor indica que es nulo. Por lo tanto, a pesar de que el transmisor sube la tasa de transmisión efectiva, el servidor/receptor le indica que durante los siguientes segundos no le puede enviar nada debido a que no da abasto y simplemente se dedica a confirmar esa enorme cantidad de tramas que el transmisor le ha enviado de golpe. Este fenómeno de que el throughput caiga drásticamente sin haber retransmisiones solo ocurre en este escenario de mayor distancia en un canal de 20 MHz con tráfico interferente.

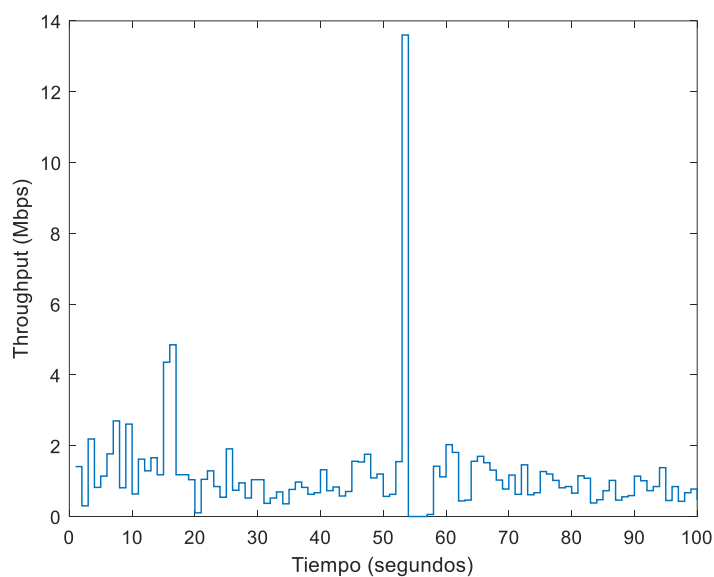
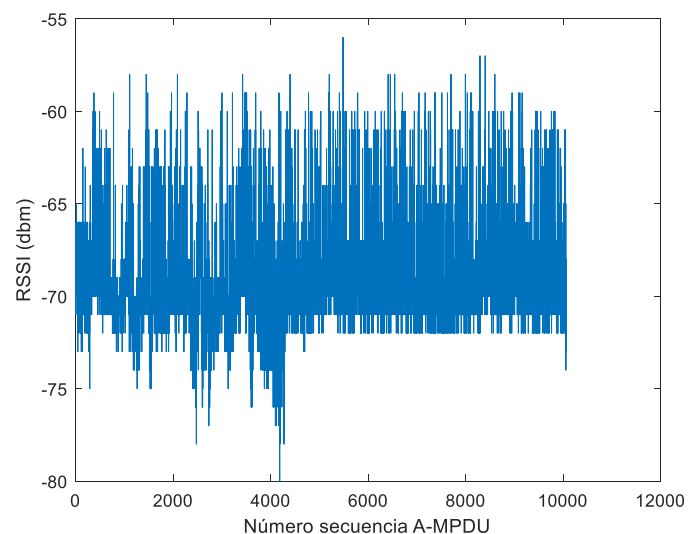
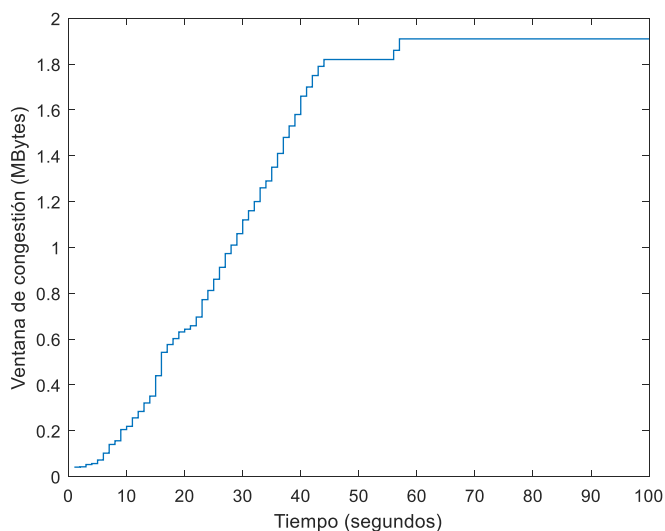
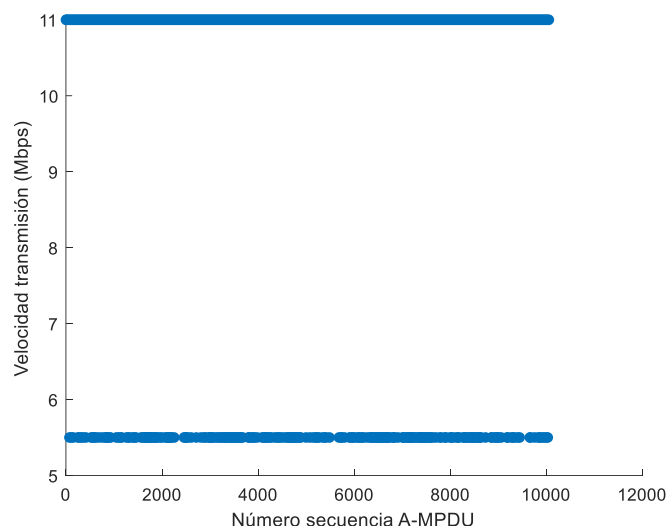
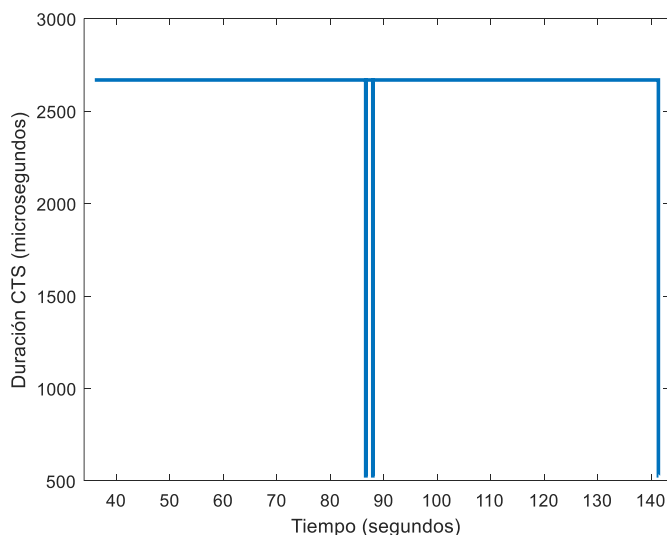


Figura 5.32. Evolución del Throughput durante la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic





Figuras 5.33. Resultados de la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando Cubic

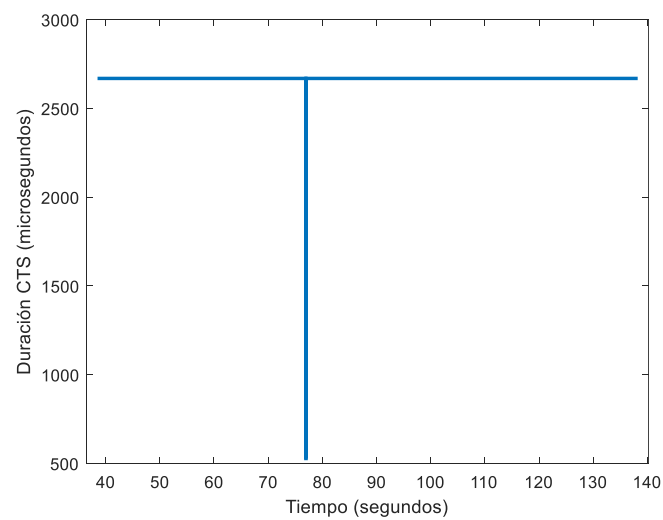
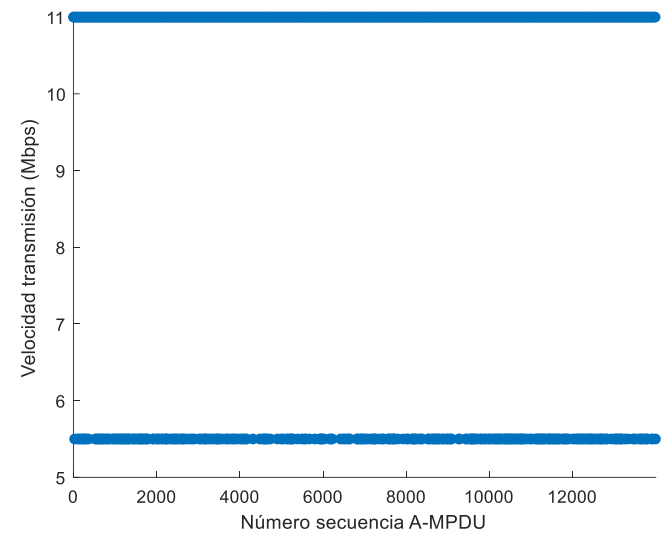
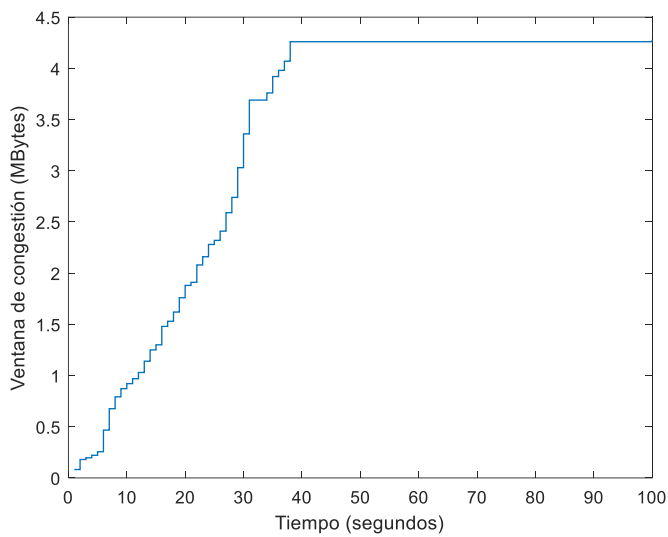
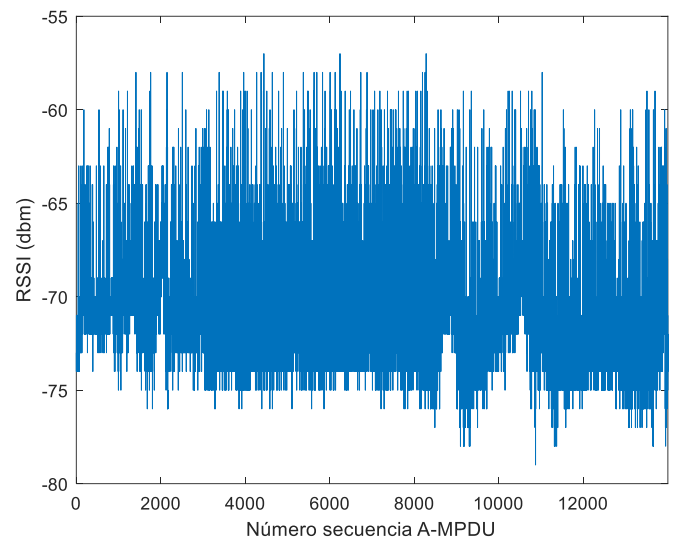
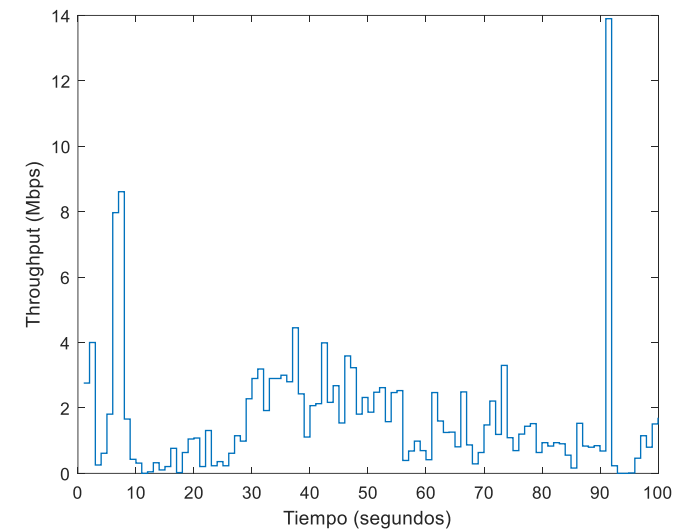
Aparte de lo comentado sobre la Figura 5.32 y la reducción del throughput de 5.27 a 1.17 Mbps, la introducción de tráfico interferente no ha provocado ningún cambio substancial para una red Wi-Fi en el escenario de mayor distancia con un canal de 20 MHz.

- Canal de 20 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia con tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 20 MHz.

Tabla 5.25. Resultados medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	1.65
Volumen datos transmitidos (MBytes)	19.7
Nº Retransmisiones	0



Figuras 5.34. Resultados de la medida del escenario de mayor distancia canal de 20 MHz con tráfico interferente utilizando New Reno

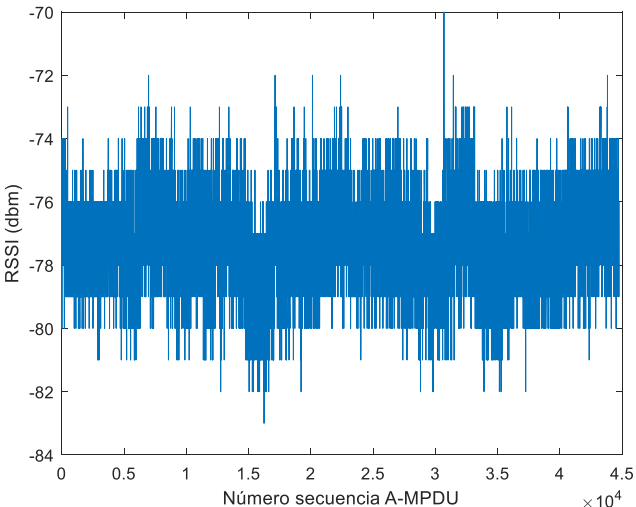
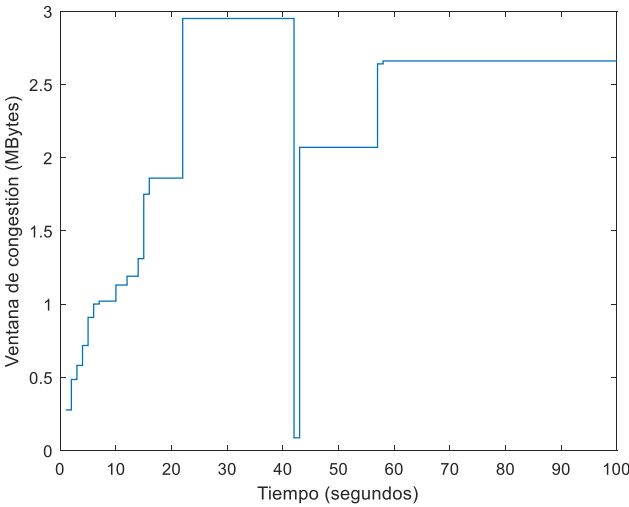
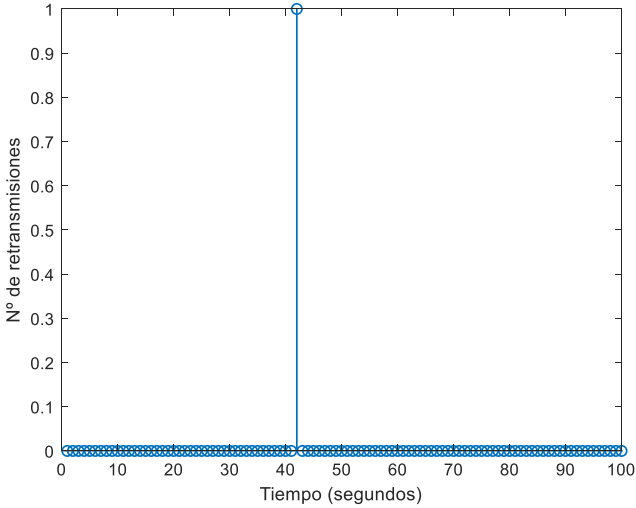
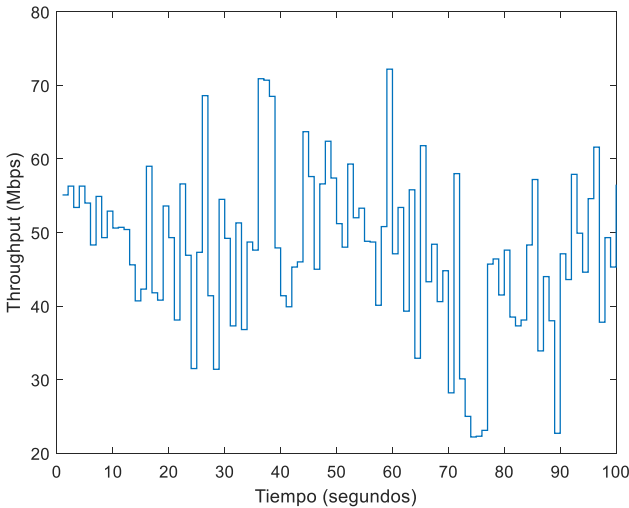
Estos resultados demuestran que utilizar New Reno o Cubic en el escenario de mayor distancia con un canal de 20 MHz con tráfico interferente no influye nada.

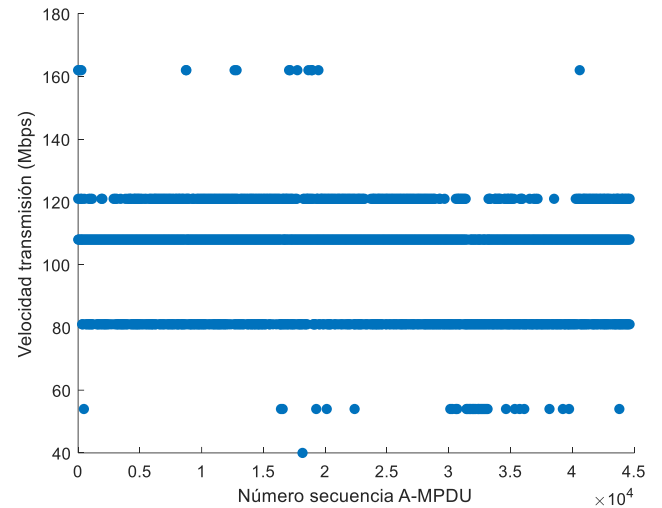
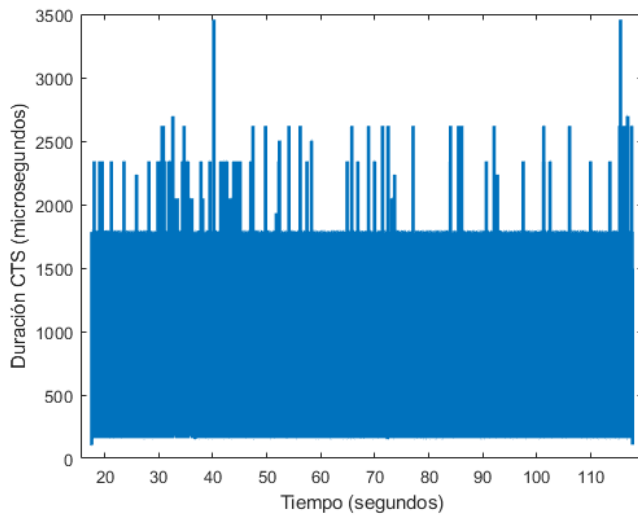
- Canal de 40 MHz, sin tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia sin tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 40 MHz.

Tabla 5.26. Resultados medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic

Throughput (Mbps)	47.5
Volumen datos transmitidos (MBytes)	567
Fast Retransmisión (Triple ACK)	1
Retransmisión por Time-Out	0





Figuras 5.35. Resultados de la medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando Cubic

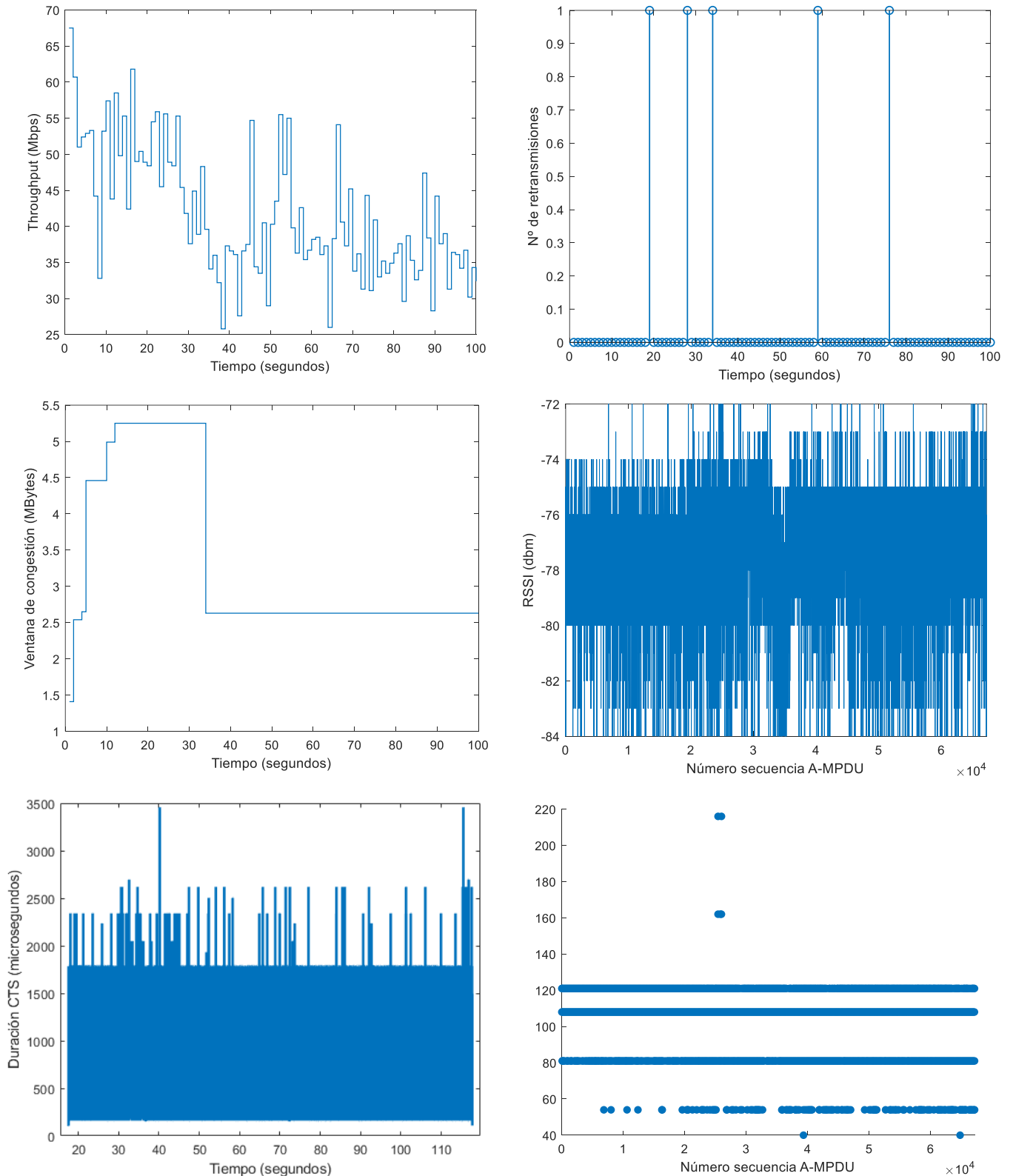
El uso de canales de 40 MHz en el escenario de mayor distancia presenta grandes ventajas. La primera de ellas es la mejora del throughput (47.5 Mbps) respecto al canal de 20 MHz. Otra es que las velocidades de transmisión rondan los 100 Mbps, y la última, es que a pesar de utilizar un canal de 40 MHz, se producen muy pocas retransmisiones TCP, habiendo en este caso una única retransmisión, que encima es una retransmisión rápida.

- Canal de 40 MHz, sin tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia sin tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 40 MHz.

Tabla 5.27. Resultados medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno

Throughput (Mbps)	41.5
Volumen datos transmitidos (MBytes)	495
Fast Retransmisión (Triple ACK)	1
Retransmisión por Time-Out	4



Figuras 5.36. Resultados de la medida del escenario de mayor distancia canal de 40 MHz sin tráfico interferente utilizando New Reno

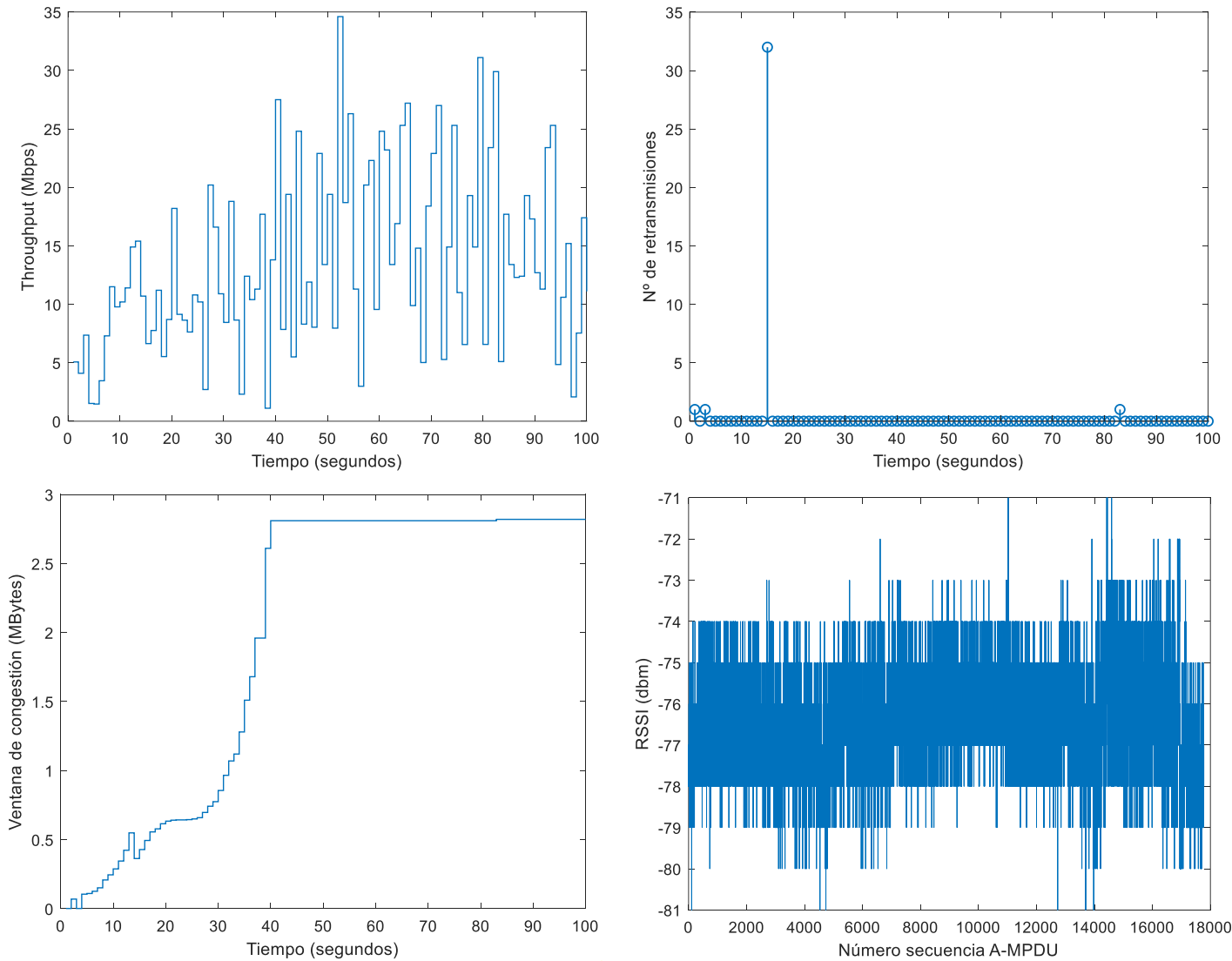
Utilizar New Reno en vez de Cubic presenta pocas diferencias y de muy poca importancia. Por ejemplo, un ligero menor throughput (41.5 Mbps) y un ligero aumento del número de retransmisiones, habiendo en este caso 5 retransmisiones TCP.

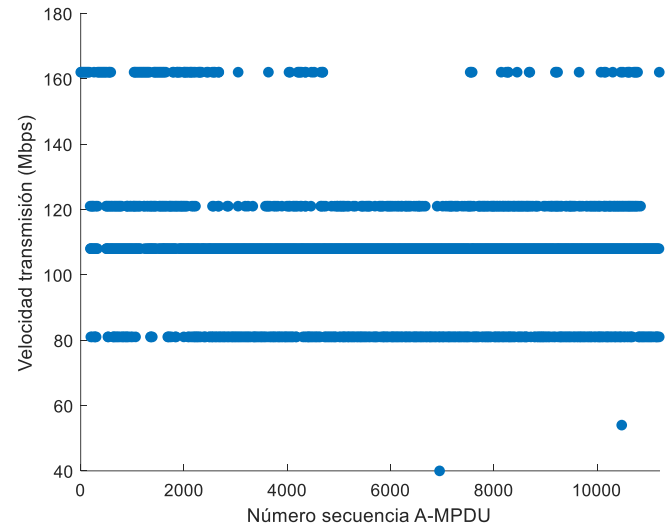
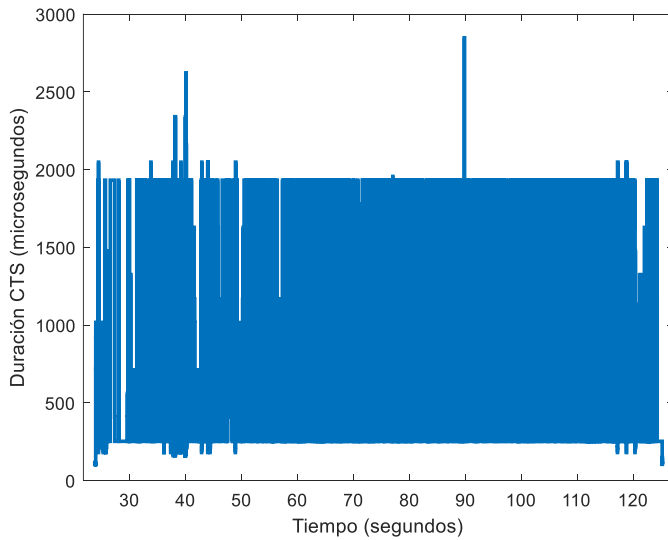
- Canal de 40 MHz, con tráfico interferente, TCP Cubic

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia con tráfico interferente empleando el algoritmo de control de congestión Cubic en un canal de 40 MHz.

Tabla 5.28. Resultados medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando Cubic

Throughput (Mbps)	13.5
Volumen datos transmitidos (MBytes)	161
Fast Retransmisión (Triple ACK)	0
Retransmisión por Time-Out	35





Figuras 5.37. Resultados de la medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando Cubic

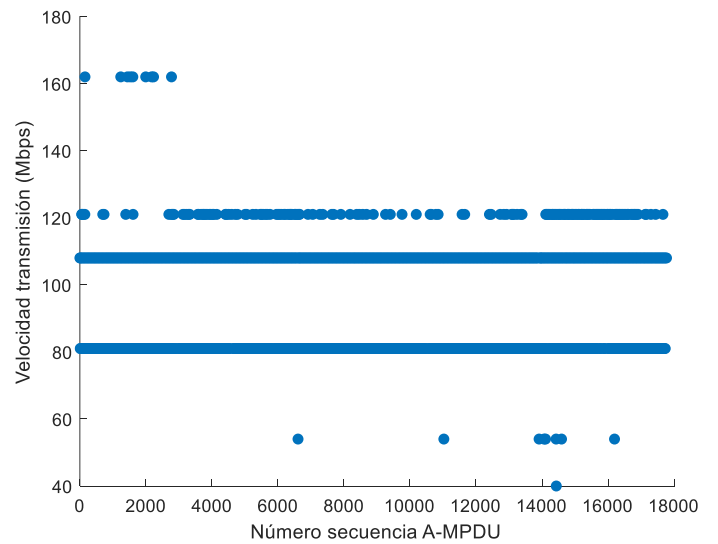
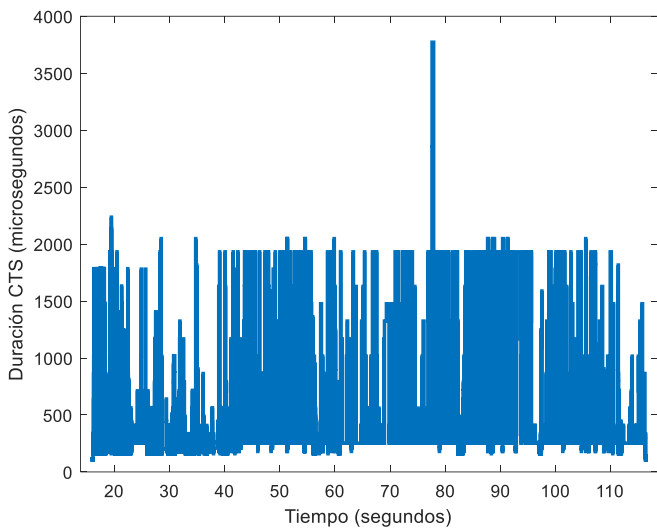
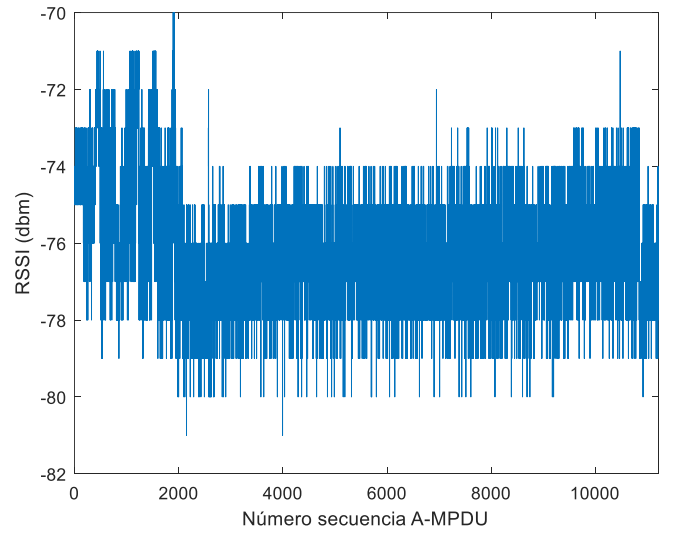
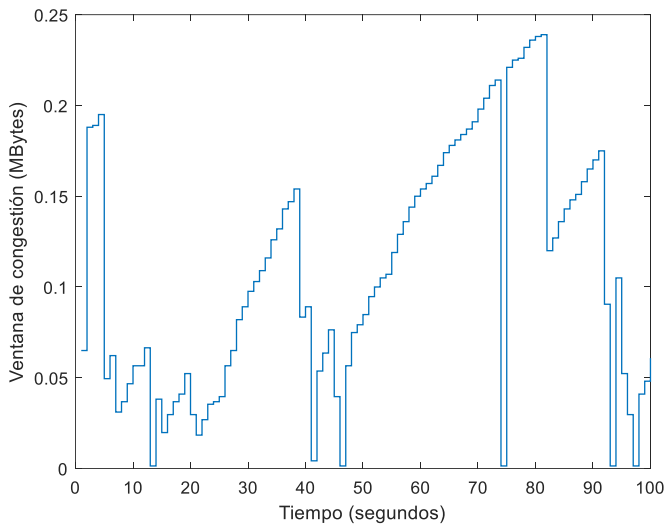
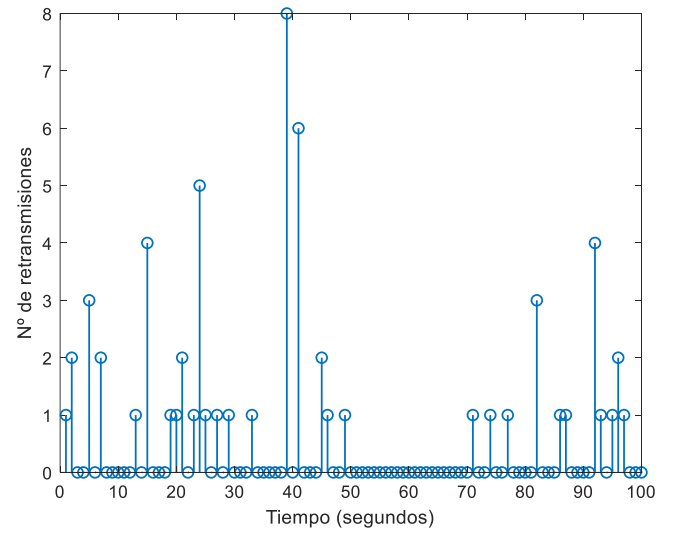
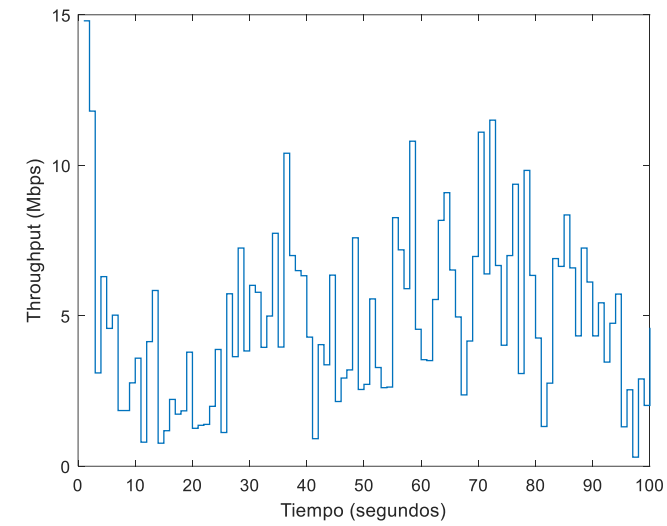
Añadir tráfico interferente a un canal de 40 MHz en el escenario de mayor distancia, al igual que en todos los casos en que se ha añadido tráfico interferente, produce una reducción del throughput a 13.5 Mbps y un aumento del número de retransmisiones, habiendo 35 retransmisiones TCP.

- Canal de 40 MHz, con tráfico interferente, New Reno

En este subapartado se mostrarán los resultados de las medidas del escenario de mayor distancia con tráfico interferente empleando el algoritmo de control de congestión New Reno en un canal de 40 MHz.

Tabla 5.29. Resultados medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando New Reno

Throughput (Mbps)	4.85
Volumen datos transmitidos (MBytes)	57.8
Fast Retransmisión (Triple ACK)	2
Retransmisión por Time-Out	60



Figuras 5.38. Resultados de la medida del escenario de mayor distancia canal de 40 MHz con tráfico interferente utilizando New Reno

Por último, el uso de New Reno en las mismas condiciones que en la medida anterior, presenta resultados ligeramente peores con un mayor número de retransmisiones TCP (62) y un descenso del throughput (4.85 Mbps).

5.4. Retransmisiones MAC

En este último apartado, se mostrarán el porcentaje de segmentos TCP que son retransmisiones a nivel de capa MAC que se han producido en cada medida en ambos sentidos, tanto TX → RX como viceversa. Este dato es proporcionado por el analizador de protocolos Wireshark. Se extraerán estos datos de los resultados de las medidas en modo monitor del transmisor.

Tabla 5.30. Porcentaje de retransmisiones a nivel de capa MAC en cada medida

Tipo Escenario	Sin tráfico Interferente				Con tráfico interferente			
	Canal de 20 MHz		Canal de 40 MHz		Canal de 20 MHz		Canal de 40 MHz	
	Cubic	New Reno	Cubic	New Reno	Cubic	New Reno	Cubic	New Reno
Menor Distancia	2.4%	1.2%	9.8%	17.5%	2.9%	1.5%	23.6%	27.9%
Distancia Media	9.4%	8.8%	33.1%	34.2%	16.5%	10.9%	31.5%	25.5%
Mayor Distancia	20.6%	20%	30.9%	35.9%	30.6%	31%	37.6%	41.3%

Se observa un importante incremento del porcentaje de retransmisiones MAC para canal de 40 MHz frente al de 20 MHz en todos los casos, y en particular, ligeramente mayor también en New Reno frente a Cubic (salvo en el caso de distancia media con tráfico interferente). Esto puede ser debido al aumento de la probabilidad de colisión en el acceso al medio al ser las tramas de menor duración en el canal de 40 MHz.

Capítulo 6. Conclusiones y líneas futuras de trabajo

En primer lugar, haciendo la comparación directa entre un canal de 20 MHz y un canal de 40 MHz, se ha mostrado claramente como es mejor en prestaciones el canal de 40 MHz. Si bien es cierto que el canal de 20 MHz es mucho más eficiente en cuestión de rendimiento por anchura de canal, siempre sería aconsejable utilizar un canal de 40 MHz, a no ser que el espectro de frecuencias de la banda 2.4 GHz esté sobre saturado. En el escenario de menor distancia, el canal de 40 MHz tiene un throughput que alcanza alrededor de los 130 Mbps, 50 Mbps más que en un canal de 20 MHz. En el escenario de distancia media, las medidas indican que el rendimiento sobre un canal de 20 MHz y un canal de 40 MHz es similar, mientras que en el escenario de mayor distancia, el rendimiento sobre un canal de 40 MHz supera ampliamente al de un canal de 20 MHz. En el escenario de mayor distancia, el throughput de un canal de 40 MHz está en torno a los 45 Mbps, mientras que en un canal de 20 MHz está en torno a los 5 Mbps, es decir, el throughput del canal de 40 MHz es 9 veces mejor que el del canal de 20 MHz, una gran mejora. Cuando hay tráfico interferente dentro de la propia red Wi-Fi, las tasas de transmisión efectiva o throughput se reducen. Esto es debido a dos razones. La primera de ellas es que al haber otras entidades transmitiendo dentro de la propia red Wi-Fi, se tienen que turnar para transmitir información y por lo tanto, los tiempos que se le otorgan al transmisor son inferiores, como se muestra en las gráficas que representan el tiempo otorgado para transmitir en las tramas CTS al transmisor. La segunda razón, es que la presencia de tráfico interferente dentro de la propia red provoca un aumento de la pérdida de paquetes, y por ende, el número de retransmisiones tanto de capa MAC como de TCP.

Antes de nada, hay que tener una cosa en cuenta y es el mecanismo de adaptación de la tasa de transmisión del chipset Atheros. El chipset de Atheros dependiendo de la potencia de señal recibida estima a qué distancia se encuentra la otra parte de la transmisión y para evitar tener muchas retransmisiones (o ninguna), adapta la tasa de transmisión. [10][11] Por ejemplo, en el escenario de menor distancia, en un canal de 20 MHz, la tasa de transmisión es de 130 Mbps utilizando el estándar 802.11n, mientras que en el escenario de mayor distancia la tasa de transmisión es de 11 Mbps y 5.5 Mbps, es decir, que incluso ni emplea el estándar 802.11n, sino que pasa a utilizar el estándar 802.11b. Lo más lógico sería pensar que al aumentar la distancia entre receptor y transmisor, el número de retransmisiones aumente, pero gracias al mecanismo de adaptación de la tasa de transmisión de Atheros, por ejemplo, en el escenario de mayor distancia, hace que no se produzca ninguna retransmisión TCP en un canal de 20 MHz, ya que al reducir tanto la tasa de transmisión acaba empleando el estándar 802.11b.

Como ya se ha explicado en el apartado anterior, las tasas de transmisión han ido variando entre escenarios. En el escenario de menor distancia se han alcanzado valores de tasa de transmisión de 130 Mbps y 300 Mbps, para un canal de 20 MHz y 40 MHz respectivamente, que son los mejores valores que se pueden conseguir en cada caso. En el escenario de distancia media, en el caso del canal de 20 MHz, la tasa de transmisión se ha mantenido principalmente en 130 Mbps, pero ya ha comenzado a haber un gran número de segmentos que se transmiten a velocidades inferiores, como 117 Mbps o 104 Mbps. En el caso del canal de 40 MHz, se nota mucho más esta reducción progresiva de la tasa de transmisión, a medida que aumenta la distancia entre transmisor y receptor. En el escenario de distancia media, el canal de 40 MHz emplea velocidades de transmisión entre los 240 y 120 Mbps, sin tener un valor claro de velocidad de transmisión. Por último, en el escenario de mayor distancia, en el canal de 20 MHz, como ya se ha mencionado en el anterior párrafo, las velocidades de transmisión bajan

drásticamente hasta los 11 Mbps y 5.5 Mbps, cambiando al estándar 802.11b. En cambio, en el canal de 40 MHz, las velocidades de transmisión se reducen a 120 Mbps, 108 Mbps y 81 Mbps, mostrando así el canal de 40 MHz una ventaja frente al canal de 20 MHz. Cabe destacar, que un canal de 40 MHz nunca podrá trabajar en el estándar 802.11b, ya que los canales de 40 MHz no se especifican en ese estándar, y que la velocidad de transmisión mínima existente sería 13.5 Mbps. Por último, hay que destacar que la presencia de tráfico interferente dentro de la propia red Wi-Fi no afecta a las velocidades de transmisión.

El número de retransmisiones TCP es más común en los canales de 40 MHz, debido a las altas velocidades de transmisión. La presencia de tráfico interferente también es una de las causas de pérdidas de paquetes TCP y su posterior retransmisión. En el canal de 20 MHz, en el escenario de menor distancia y en el escenario de mayor distancia no presentan casi pérdidas de paquetes, en el primero debido a la buena calidad del enlace al estar al lado, y en el segundo debido a las bajas velocidades de transmisión que utiliza el transmisor a causa del mecanismo de adaptación de velocidades de transmisión de Atheros. Como ya se mencionó, hay dos tipos de retransmisión, por recibir un triple ACK idéntico, método por el cual el receptor indica que no se ha recibido un segmento, también llamado “Fast Retransmission (Retransmisión rápida)” y una retransmisión por time-out, que ocurre cuando el transmisor no ha recibido la confirmación ACK de un segmento en un periodo de tiempo. Las retransmisiones no solamente producen una reducción del throughput, ya que hay que volver a enviar un mismo segmento, sino que también una reducción de la ventana de congestión, provocando que el transmisor esté limitado en la cantidad de datos que pueda enviar. La ventana de congestión se recupera rápidamente de las retransmisiones rápidas mientras que las retransmisiones por time-out provocan una mayor reducción de la ventana de congestión. Uno de los problemas de iperf3, es que te da los resultados de la medida cada segundo, por lo que en algunos casos, como en Cubic, que tiene unos crecimientos exponenciales tras perder un segmento, aunque se indique que haya una retransmisión, hay veces que indica que se ha mantenido o incluso en raros casos crecido. Esto se produce porque en ese mismo segundo, a la ventana de congestión le da tiempo de reducir su valor y recuperar el valor que tenía previamente a la retransmisión en ese mismo segundo. Esto también se reflejado en la gráfica que muestra la evolución del throughput a lo largo de la medida. Donde más claramente se ve la caída de la ventana de congestión y del throughput es cuando se produce una ráfaga de retransmisiones por time-out. Superponiendo las gráficas relacionadas con la evolución del throughput, el número de retransmisiones en cada instante de tiempo y la evolución de la ventana de congestión se puede ver claramente como las retransmisiones afectan tanto al throughput como a la ventana de congestión.

Por otra parte, no se ha observado una gran dependencia al algoritmo de control de congestión utilizado. El utilizar New Reno o Cubic como algoritmo de control de congestión TCP no ha influido mucho. Los valores de throughput obtenidos son bastante similares. La principal diferencia ocurre en la ventana de congestión. El algoritmo TCP Cubic por su naturaleza presenta incrementos muy rápidos en cortos instantes y bajadas bruscas hasta mantenerse estable en un punto. Mientras, New Reno suele tener un crecimiento más gradual y no tan drástico. Donde se puede observar una mayor diferencia entre New Reno y Cubic, es en las gráficas de la evolución de la ventana de congestión durante el tiempo de medida del escenario de distancia media, con especial énfasis a las medidas en un canal de 40 MHz. Otro dato diferencial entre New Reno y Cubic, que sería como resultado de los incrementos tan rápidos y drásticos de la ventana de congestión de Cubic, es que las medidas que utilizan Cubic presentan un mayor número de retransmisiones TCP, excepto en el escenario de mayor distancia, donde utilizar New Reno produce un mayor número de retransmisiones TCP.

Como líneas de trabajo futuro posible, hay varias opciones. La primera de ellas utilizando todavía el estándar 802.11n, consiste en trabajar en el espectro de frecuencias de los 5 GHz, el cual está menos ocupado. En este trabajo no se han podido realizar pruebas dicha banda porque el adaptador USB Wi-Fi utilizado no lo permite. Otra posible línea de trabajo utilizando el estándar 802.11n sería utilizar un nuevo adaptador USB Wi-Fi más potente, que soporte más spatial streams, y realizar de nuevo estas medidas pero con tasas de transmisión mucho más altas. Con 3 spatial streams en un canal de 20 MHz, en el estándar 802.11n, se puede alcanzar una velocidad de transmisión de 216.7 Mbps, y en un canal de 40 MHz, una velocidad de transmisión de hasta 450 Mbps (MCS Index 23). Por último, con el estándar 802.11n, se podría comprobar la eficacia de la técnica “Beamforming” y ver en cuánto mejoran los resultados.

Otra posibilidad sería la utilización de otro estándar posterior al estándar 802.11n, como por ejemplo, el estándar 802.11ac. Este estándar presenta múltiples mejoras. La primera sería una mejor velocidad de transmisión incluso en las mismas condiciones que en el estándar 802.11n. Es decir, si por ejemplo este trabajo se hubiera realizado con el estándar 802.11ac, y se tuviera los mismos spatial streams que se tienen ahora, 2, y la misma anchura de canal, 20 y 40 MHz, se podría conseguir velocidades de transmisión de 173.3 Mbps para un canal de 20 MHz y de 400 Mbps para un canal de 40 MHz. El estándar 802.11ac, no es solo mejor que el estándar 802.11n si utiliza sus mismas prestaciones, sino que añade novedades como por ejemplo canales con anchura de 80 MHz y 160 MHz. La única ventaja que presenta el estándar 802.11n frente al estándar 802.11ac, es que puede trabajar tanto en la banda de los 2.4 GHz como en la banda de los 5 GHz, mientras que el estándar 802.11ac solo puede trabajar en la banda de los 5 GHz.

Por último, se podría realizar este mismo trabajo pero probando el rendimiento y eficacia de otros algoritmos de control de congestión TCP. En este caso se han utilizado New Reno y Cubic que son los más comunes, pero hay mucho otros como TCP Illinois, Binary Increase Congestion (BIC), TCP Westwood+, H-TCP, TCP Vegas, High Speed TCP, TCP-Hybla y YeAH TCP, entre otros.

En definitiva, todavía queda un amplio margen para la caracterización de los estándares IEEE 802.11, y sobre todo con el despliegue de las últimas especificaciones como son la 802.11ax, también conocido como Wi-Fi 6, y la futura especificación 802.11be o Wi-Fi 7 cuya ratificación por parte del IEEE está prevista para el 2024. [12]

Bibliografía

- [1] Apuntes asignatura “Redes inalámbricas” de la Universidad de Cantabria
- [2] R. Karmakar, S. Chattopadhyay, S. Chakraborty, “Impact of IEEE 802.11n/ac PHY/MAC High Throughput Enhancements on Transport and Application Protocols – A Survey”, *IEEE Communications Surveys and Tutorials*, DOI 10.1109/COMST.2017.2745052, 2017
- [3] «IEEE 802.11 Block Acknowledgment — INET 4.3.0 documentation». <https://inet.omnetpp.org/docs/showcases/wireless/blockack/doc/index.html> (accedido jul. 16, 2021).
- [4] C. Marathe, «Understanding A-MPDU & Block Ack through Wireless captures», *WiFi Wiki*, nov. 19, 2019. <https://wifiwiki.wordpress.com/2019/11/19/understanding-a-mpdu-block-ack-through-wireless-captures/> (accedido jul. 16, 2021).
- [5] 80211 80211, «BLOCK ACK». <http://www.netprojnetworks.com/block-ack/> (accedido jul. 16, 2021).
- [6] «What is Transmission Control Protocol (TCP) | SDNCentral», *SDxCentral*. <https://www.sdxcentral.com/resources/glossary/transmission-control-protocol-tcp/> (accedido jun. 07, 2021)
- [7] Apuntes asignatura “Diseño y Operación de Redes Telemáticas” de la Universidad de Cantabria
- [8] S.Floyd, T.Henderson “The NewReno Modification to TCP’s Fast Recovery Algorithm” RFC 2582, Apr 1999.
- [9] E. Lopez-Aguilera, E. Garcia-Villegas, y J. Casademont, «Evaluation of IEEE 802.11 coexistence in WLAN deployments», *Wireless Netw*, vol. 25, n.º 1, pp. 87-104, ene. 2019, doi: 10.1007/s11276-017-1540-z.
- [10] R. K. Sheshadri and D. Koutsonikolas, "On packet loss rates in modern 802.11 networks," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, 2017,pp. 1-9. doi: 10.1109/INFOCOM.2017.8057130
- [11] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, Kevin Almeroth, A practical framework for 802.11 MIMO rate adaptation, *Computer Networks*, Volume 83, 2015, Pages 332-348, ISSN 1389-1286, <https://doi.org/10.1016/j.comnet.2015.03.015>.
- [12] A. Garcia-Rodriguez, D. Lopez-Perez, L. Galati-Giordano, y G. Geraci, «IEEE 802.11be: Wi-Fi 7 Strikes Back», *IEEE Commun. Mag.*, vol. 59, n.º 4, pp. 102-108, abr. 2021, doi: 10.1109/MCOM.001.2000711.

Anexo A. Código C “ProcesaTCP.c”

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  void main(){
6
7  FILE *myFile, *myFile2, *myFile3;
8  myFile = fopen("TCP.txt", "r");
9
10     char input[300];
11     char listaId[50];
12     char TipoTCP1[100];
13     char TipoTCP2[100];
14     char TCPRTX []="Retransmission";
15     char TCPOut []="Out-Of-Order";
16     int pot[80000];
17     int i, j, s, longitud, Antena, ratecont, id, contrama;
18     float rate[80000];
19     Antena=-1;
20     ratecont=-1; //inicializo en -1 para que no salte ciclos
21     id=-1;
22     j=0;
23     s=1;
24     i=0;
25     contrama=0;
26
27     //Lectura
28
29     while(fscanf(myFile, " %[^\\n]", input) != EOF){
30
31         if (i==s){
32
33             sscanf(input,"%s %s %s %s %s %i %s %s",&longitud , TipoTCP1, TipoTCP2)
34
35             if(longitud == 1579){
36
37                 ratecont = s+8;
38
39             }else if(longitud == 1574){
40
41                 ratecont=s+6;
42
43             } else{
44
45                 id = s + 20;
46
47             }
48         }
49
50         if (i == ratecont){
51
52             Antena = ratecont +3;
53             sscanf(input,"%s %s %f",&rate[j]);
54
55         }
```

```

56
57 if (i == Antena){
58
59     sscanf(input,"%*s %*s %*s %i",&pot[j]);
60     id= Antena + 10;
61     j = j+1;
62
63 }
64
65 if(i == id){
66
67     sscanf(input,"%*s %*s %s",listaid);
68     contrama= contrama +1;
69
70     if(strcmp(TCPOut,TipoTCP2) == 0 || strcmp(TCPRTX,TipoTCP2) == 0){
71
72         s=id+11;
73
74     }else{
75
76         s = id + 103;
77
78     }
79
80 }
81
82 i = i + 1;
83
84 }
85
86 fclose(myFile);
87
88 //Escritura
89
90 myFile2 = fopen("SolSen.txt", "w");
91
92 for(i=0;i<j;i++){
93
94     fprintf(myFile2,"%i\n",pot[i]);
95
96 }
97
98 fclose(myFile2);
99
100 myFile3 = fopen("SolRate.txt", "w");
101
102 for(i=0;i<j;i++){
103
104     fprintf(myFile3,"%f\n",rate[i]);
105
106 }
107
108 fclose(myFile3);
109
110 }

```

Anexo B. Código C “ProcesaCTS.c”

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  void main(){
6
7  FILE *myFile, *myFile2;
8  myFile = fopen("CTS.txt", "r");
9
10     char input[300];
11     int i, j, s, Cts;
12     float TiempoTrama[80000];
13     int CtsTiempo[80000];
14     Cts=-1;
15     j=0;
16     s=1;
17     i=0;
18
19     while(fscanf(myFile, " %[^\\n]", input) != EOF){
20
21         if (i==s){
22
23             sscanf(input,"%*s %f",&TiempoTrama[j]);
24             Cts= s+7;
25
26         }
27
28         if (i == Cts){
29
30             sscanf(input,"%*s %*s %*s %*s %*s %*s %i",&CtsTiempo[j]);
31             s= Cts + 5;
32             j=j+1;
33
34         }
35
36         i = i + 1;
37     }
38
39     fclose(myFile);
40
41     //Escritura
42
43     myFile2 = fopen("CTSprocesado.txt", "w");
44
45     for(i=0;i<j;i++){
46
47         fprintf(myFile2,"%f %i\\n",TiempoTrama[i], CtsTiempo[i]);
48
49     }
50
51     fclose(myFile2);
52
53 }
```

Anexo C. Código Matlab “dibuja.m”

```
1 function dibuja(x, y)
2
3     fileID = fopen(x,'r');
4     formatSpec = '%d';
5     A = fscanf(fileID,formatSpec);
6     LimSup=size(A);
7
8     figure(1)
9     plot(A)
10    xlim([0 LimSup(1)]);
11    ylabel('RSSI (dbm)')
12    xlabel('Número secuencia A-MPDU')
13    fclose(fileID);
14
15    fileID2 = fopen(y,'r');
16    formatSpec2 = '%f';
17    B = fscanf(fileID2,formatSpec2);
18    Tam=size(B);
19    Y=1:1:Tam;
20
21    figure(2)
22    scatter(Y,B,'filled')
23    xlim([0 LimSup(1)])
24    ylabel('Velocidad transmisión (Mbps)')
25    xlabel('Número secuencia A-MPDU')
26    fclose(fileID2);
27
28 end
```


Anexo D. Código Matlab “dibujaCTS.m”

```
1  function dibujaCTS(x)
2
3  -   fileID = fopen(x,'r');
4  -   formatSpec = '%f %d';
5  -   sizeA = [2 Inf];
6  -   A = fscanf(fileID,formatSpec,sizeA);
7  -   A=A';
8  -   Fin=size(A);
9
10 -   figure(1)
11 -   LimInf=A(1,1)-2;
12 -   LimSup=A(Fin(1),1)+2;
13 -   stairs(A(:,1), A(:,2), 'LineWidth', 2)
14 -   ylabel('Duración CTS (microsegundos)')
15 -   xlabel('Tiempo (segundos)')
16 -   xlim([LimInf LimSup])
17 -   fclose(fileID);
18
19 - end
```